Uniwersytet Śląski w Katowicach
Wydział Informatyki i Nauki o Materiałach
Informatyka

Rozprawa doktorska

Wydobywanie wiedzy z danych złożonych

mgr Tomasz Xięski

Promotor: prof. dr hab. inż. Alicja Wakulicz-Deja
Promotor pomocniczy: dr Agnieszka Nowak-Brzezińska

Katowice, 2014

Kochanym Rodzicom, Siostrze, Szwagrowi i Lucynie.

Wyrażam zgodę na udostępnienie mojej pracy doktorskiej dla celów naukowo-badawczych.

Data:

Podpis autora:

Słowa kluczowe: *analiza skupień, wydobywanie wiedzy, dane złożone, DBSCAN, OPTICS, AHC, algorytmy gęstościowe, wizualizacja skupień.*

Oświadczenie autora pracy

Świadomy odpowiedzialności prawnej oświadczam, że niniejsza praca doktorska została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data Podpis autora pracy

Spis treści

Podziękowania

Pragnę serdecznie podziękować mojemu promotorowi Profesor Alicji Wakulicz-Deji za wszechstronną pomoc, cenne uwagi i opiekę podczas powstawania niniejszej pracy. Podobnie chciałbym podziękować promotorowi pomocniczemu Doktor Agnieszce Nowak-Brzezińskiej za wielogodzinne konsultacje, okazaną pomoc i troskę na każdym etapie prowadzonych badań. Bez Ich udziału i zaangażowania w pracę naukowo-badawczą autora, niniejsza praca z całą pewnością by nie powstała.

Dziękuję moim najbliższym, za wyrozumiałość, cierpliwość i nieocenione wsparcie, które pomogły mi podczas trudów pisania pracy.

Wszystkim osobom, którzy wpierali mnie w czasie pisania niniejszej rozprawy, a w szczególności Kolegom i Koleżankom z Zakładu Systemów Informatycznych UŚ. Bardzo Wam dziękuję.

Rozdział 1

Wprowadzenie

Nieustanny rozwój techniki oraz rosnące możliwości sprzętu komputerowego umożliwiają przechowywanie bardzo dużych ilości danych we wszelkiego rodzaju bazach i repozytoriach. Dane te najczęściej zbierane są w sposób automatyczny, wykorzystując szereg czujników lub systemów monitorujących. Nawet niewielkie transakcje w sklepie dokonywane kartą kredytową czy rozmowy telefoniczne są rejestrowane przez komputery. Zwykle wiele parametrów jest zapisywanych jednocześnie, co skutkuje wysoką wymiarowością zbioru danych. Dane te są gromadzone, ponieważ zakłada się, że mogą być źródłem nieznanych, potencjalnie użytecznych wzorców, korelacji i trendów. Odkryte wzorce, wyrażone w postaci modelu analitycznego, mogą posiadać skomplikowaną strukturę, przez co są trudne do dalszej analizy. Jednakże to nie tylko nadmierna ilość danych wpływa na trudności badawcze. Istotniejszym czynnikiem jest ich złożona struktura zarówno pod względem dużej liczby atrybutów opisujących każdy obiekt danych, jak również użytych typów danych. Informacje zakodowane w bazie często opisane są atrybutami różnych typów, wliczając w to wartości binarne, dyskretne, ciągłe, kategoryczne, tekstowe czy reprezentujące daty. Tego typu dane można nazwać złożonymi i będą one podstawą analizy w niniejszej rozprawie. Wnikliwa analiza tematyki związanej z niniejszą rozprawą, a także wyniki uzyskane w ramach przeprowadzonych badań pozwoliły uformować tezę pracy, zgodnie z którą:

> *Opracowanie struktury dla złożonych baz wiedzy oraz procesu wyszukiwania umożliwi efektywne wydobywanie wiedzy z rzeczywistych zbiorów danych złożonych.*

Celem rozprawy jest zatem opracowanie metody wydobywania wiedzy ze złożonych zbiorów danych rzeczywistych o dużej liczebności, uwzględniającej ich specyfikę i dziedzinowy charakter oraz efektywne środki wizualizacji wydobytej wiedzy. Mianem wiedzy, na potrzeby niniejszej pracy, określa się wyrażoną w postaci wzorców, trendów czy korelacji "informację odnośnie otaczającego świata, która umożliwia ekspertowi podejmowanie decyzji" [6]. Badania zostaną oparte na dwóch rzeczywistych zbiorach: pierwszy z nich zawiera dane dotyczące funkcjonowania urządzeń nadawczo-odbiorczych operatora telefonii komórkowej (rozmieszczonych na te-

renie aglomeracji śląskiej), drugi agreguje statystyki gromadzone w oprogramowaniu AirSync, związane z zarządzaniem sieciami bezprzewodowymi. Mimo, że oba zestawy danych wydają się być ze sobą mocno powiązane pod względem tematyki telekomunikacyjnej, to jednak posiadają zupełnie odmienną strukturę i charakterystykę.

Spośród wielu technik eksploracji danych zdecydowano się wybrać analizę skupień [7] i to właśnie wszystkie aspekty realizacji tej techniki w odniesieniu do danych złożonych są podstawą niniejszej rozprawy. Wydobywanie wiedzy z rzeczywistych baz wiedzy jest procesem wieloetapowym i stawia szereg wymogów wobec algorytmów grupowania jak: możliwość odkrywania skupień o różnej strukturze, odporność na występowanie wartości izolowanych, posiadanie relatywnie niskiej złożoności obliczeniowej i zajętości pamięci, jasno określone kryteria stopu algorytmu oraz wysoka jakość tworzonych skupień. Niestety klasyczne metody analizy skupień (jak niehierarchiczny algorytm k-średnich) nie spełniają podanych wymagań. Dodatkowo takie rzeczywiste bazy danych najczęściej charakteryzują się występowaniem wartości pustych (brakujących), niezdyskretyzowanych, czy zduplikowanych, co znacząco utrudnia ich przetwarzanie oraz analizę. Zatem w procesie badawczym wykorzystywane są bardziej złożone algorytmy, ale te dla osiągnięcia optymalnego rezultatu wymagają zdefiniowania różnej liczby parametrów. Dlatego też istotnym problemem, omawianym w dalszej części pracy, jest określenie metod: optymalnego doboru parametrów dla procesu grupowania oraz opisu utworzonej struktury złożonych grup.

Niniejsza praca odnosi się również do problemu, w jaki sposób wizualizacja danych może funkcjonować jako efektywne i autonomiczne narzędzie analizy, jak również służyć jako technika łącząca wiedzę dziedzinową i zdolności kognitywne człowieka w procesie odkrywania wiedzy. Omawia proces graficznej analizy eksploracyjnej (ang. *visual data mining*) [3] oraz dokonuje porównania najpopularniejszych technik prezentacji skupień, spotykanych w literaturze przedmiotu, z autorską koncepcją opartą na algorytmie generowania tzw. map prostokątów (ang. *treemaps*) [11]. Kolejnym istotnym aspektem omawianym w rozprawie jest przegląd i porównanie możliwości obecnie dostępnych systemów analizy danych, wraz z wykazaniem ich wad i zalet, szczególnie w kontekście efektywności zaimplementowanych technik grupowania. Stanowi to jednocześnie motywację do stworzenia autorskiego systemu wydobywania wiedzy *DensGroup*.

1.1 Układ pracy

Rozdział drugi wprowadza niezbędne pojęcia związane z dziedziną telekomunikacji oraz akcentuje znaczenie wiedzy dziedzinowej w procesie analizy i przetwarzania rzeczywistych zbiorów złożonych danych. Omawia strukturę i wyjaśnia różnice między badanymi zestawami danych jak również prezentuje koncepcję wydobywania wiedzy (proponowaną przez autora pracy), dostosowaną do ich złożonej postaci.

Trzeci rozdział poświęcony jest natomiast porównaniu dostępnych rozwiązań programowych stosowanych do wydobywania wiedzy z danych. Prezentuje wady i zalety omówionych systemów jak również powody dla których został stworzony autorski system analizy *DensGroup*.

Kolejny rozdział poświęcony jest metodom opisu danych, z uwzględnieniem prostych sta-

tystyk (zarówno centralnej tendencji jak i rozproszenia) oraz technik graficznych. Przedstawia również przyczyny wyboru konkretnej reprezentacji opisowo–eksploracyjnej skupień, mającej zastosowanie w narzędziu *DensGroup*.

Celem rozdziału piątego jest omówienie porównawcze gęstościowych algorytmów analizy skupień oraz potwierdzenie ich użyteczności w kontekście grupowania rzeczywistych zbiorów danych złożonych. Wyszczególniono cechy odróżniające tę grupę technik względem podejść klasycznych (zarówno hierarchicznych jak i podziałowych), a ponadto przedstawiono najważniejsze aspekty związane z budową i implementacją tychże algorytmów.

W rozdziale szóstym znajduje się omówienie procesu graficznej analizy eksploracyjnej oraz porównanie najpopularniejszych graficznych technik reprezentacji skupień, spotykanych w literaturze przedmiotu, z autorską koncepcją opartą o algorytm generowania tzw. map prostokątów. Analiza możliwości omówionych technik oparta była na rzeczywistych przykładach, prezentujących zajętość plików i katalogów na dysku twardym.

Treścią rozdziału siódmego jest dokumentacja autorskiego systemu wydobywania wiedzy *DensGroup*. Przestawiono zarówno proces instalacji i obsługi wspomnianego oprogramowania, jak również pokazano rzeczywisty scenariusz zastosowania tegoż systemu (odróżniający go od innych tego typu) na jednym, z omawianych w rozdziale drugim zestawów danych. Eksperymenty przeprowadzone na potrzeby niniejszej rozprawy przedstawione zostały w kolejnym rozdziale (ósmym).

Pracę kończy podsumowanie, w którym starano się potwierdzić i skonfrontować tezę postawioną na początku pracy, z rezultatami uzyskanymi w trakcie prowadzonych badań. Jako dodatek dołączono również słownik najważniejszych pojęć wykorzystywanych w niniejszej rozprawie.

Rozdział 2

Problematyka analizy zbiorów rzeczywistych

Wydobywanie wiedzy ukrytej w danych stało się szczególnie istotne w ostatnich latach, gdy mamy do czynienia z nieustannie rosnącą liczbą informacji przechowywanych w bazach i hurtowniach danych. Problemem staje się nie samo gromadzenie danych, a ich skuteczna analiza, odkrywanie korelacji, trendów, czy wyciąganie poprawnych wniosków. Skala problemu jest wyraźnie widoczna w odniesieniu do baz agregujących dane telekomunikacyjne (będących przedmiotem badań niniejszej rozprawy). Według Cisco Virtual Networking Index [79], transfer danych generowany przez urządzenia mobilne w 2012 roku wyniósł około 885 petabajtów miesięcznie (co stanowi dwudziestoprocentowy wzrost względem roku poprzedniego). Ponadto szacuje się, że do 2017 roku parametr ten osiągnie ponad 11 eksabajtów [79] dzięki rosnącej popularności rozbudowanych telefonów (ang. *smartphones*). Warto również nadmienić, że przekroczenie granicy zetabajta przez globalny ruch (transfer) sieciowy przewidywane jest już za dwa lata.

Wzrost zapotrzebowania na dostęp do globalnych informacji, rozrost sieci społecznościowych oraz malejące ceny sprzętu elektronicznego stały się motorem napędowym rozwoju nauki i techniki, zwłaszcza w kategorii szerokopasmowych sieci bezprzewodowych oraz telekomunikacyjnych. W celu zaspokojenia rosnących wymagań użytkowników wprowadzane są coraz to nowsze usługi i pakiety transmisji danych, w tym dostęp do cyfrowej telewizji czy wieloosobowych wideokonferencji. Usługi IT (ang. ***I****nformation* ***T****echnology*) automatycznie dostosowują się obecnie do zmiennych potrzeb biznesu, a to wymaga bardziej efektywnego zarządzania siecią, zwłaszcza w dużych przedsiębiorstwach i korporacjach. Niestety monitorowanie i utrzymanie skomplikowanych sieci telekomunikacyjnych jest zadaniem trudnym, przez co również często kosztownym. Problem stanowią nie tylko zmieniające się wymogi użytkowników, czy nierównomierny rozkład obciążenia sieci, ale również niekompatybilność urządzeń sieciowych względem siebie. Dlatego też obserwuje się znaczący wzrost zapotrzebowania na oprogramowanie, które potrafi automatycznie zbierać i przetwarzać dane pochodzące z różnych źródeł

i urządzeń sieciowych oraz na tej podstawie odpowiednio reagować np. przez wysłanie monitu do administratora. Przykładem takiego oprogramowania jest system AirSync [90] do zarządzania sieciami bezprzewodowymi. Jednakże oprócz informacji o awarii, istotniejszym czynnikiem z punktu widzenia usługodawcy jest przyczyna powstałego problemu. W obliczu natłoku zgromadzonych danych, przechowywanych często w różnych źródłach i formatach, nie jest możliwa ich dogłębna analiza (celem wykrycia przyczyn problemów) przy wykorzystaniu tradycyjnych technik statystycznych. Dlatego też poszukuje się metod generujących zależności, trendy, czy relacje, które mogą zostać zaaplikowane do dużych zbiorów danych, celem wygenerowania nowej i poprawnej wiedzy (np. na temat pracy urządzeń czy awarii sieci).

Jednak to nie tylko nadmierna liczba danych wpływa na trudność ich analizy. Ważniejszym czynnikiem jest ich złożona struktura zarówno pod względem dużej liczby atrybutów opisujących każdy obiekt danych, jak również użytych typów danych. Informacje zakodowane w bazie często opisane są atrybutami różnych typów, wliczając w to wartości binarne, dyskretne, ciągłe, kategoryczne, tekstowe czy reprezentujące daty. Tego typu dane można nazwać złożonymi i będą one podstawą analizy w niniejszej rozprawie.

2.1 Rola wiedzy dziedzinowej w procesie odkrywania wiedzy

W przedstawionej rozprawie dwa rzeczywiste zbiory danych złożonych zostaną poddane badaniom. Pierwszy zbiór zawiera dane dotyczące funkcjonowania urządzeń nadawczo-odbiorczych operatora telefonii komórkowej (rozmieszczonych na terenie aglomeracji śląskiej), drugi agreguje statystyki gromadzone w oprogramowaniu AirSync, związane z zarządzaniem sieciami bezprzewodowymi. Mimo, że oba zestawy danych wydają się być ze sobą mocno powiązane pod względem tematyki telekomunikacyjnej, to jednak posiadają zupełnie odmienną strukturę i charakterystykę, co w znaczący sposób wpływa na przebieg dalszej analizy eksploracyjnej. Jednakże przed przystąpieniem do szczegółowego omówienia wspomnianych zbiorów, należy zaznaczyć wpływ i rolę wiedzy dziedzinowej, nie tylko w procesie rewizji, interpretacji i oceny uzyskanych ostatecznie rezultatów, lecz również jako czynnik, który ma znaczący wpływ na przebieg samej analizy eksploracyjnej. Analityk danych może, na podstawie wiedzy dziedzinowej, jeszcze przed zastosowaniem konkretnego narzędzia eksploracji (np. analizy skupień), wykryć obserwacje odstające czy błędne i następnie na nie skierować swoją uwagę, może też zmodyfikować początkowy cel oraz zakres badań. Ponadto w pewnych przypadkach wyniki wstępnych eksperymentów mogą również być wykorzystywane do modyfikacji początkowej struktury zestawu danych. Przykładowo na podstawie eksperymentów autora przeprowadzonych w pracy [72], dotyczących grupowania urządzeń nadawczo-odbiorczych, dokonano obserwacji pozwalającej na redukcję liczby obiektów, poprzez wprowadzenie niewielkich zmian w bazie danych (co zostanie opisane w dalszej części tego rozdziału). Jednakże wspomniana redukcja była możliwa dopiero po zdobyciu stosownej wiedzy na temat telekomunikacji. Dlatego też koniecznym jest wprowadzenie pewnych pojęć i charakterystyk związanych z funkcjonowaniem telefonii komórkowej.

Systemy telefonii komórkowej od innych bezprzewodowych systemów łączności radiowej odróżniają dwie istotne cechy [13]:

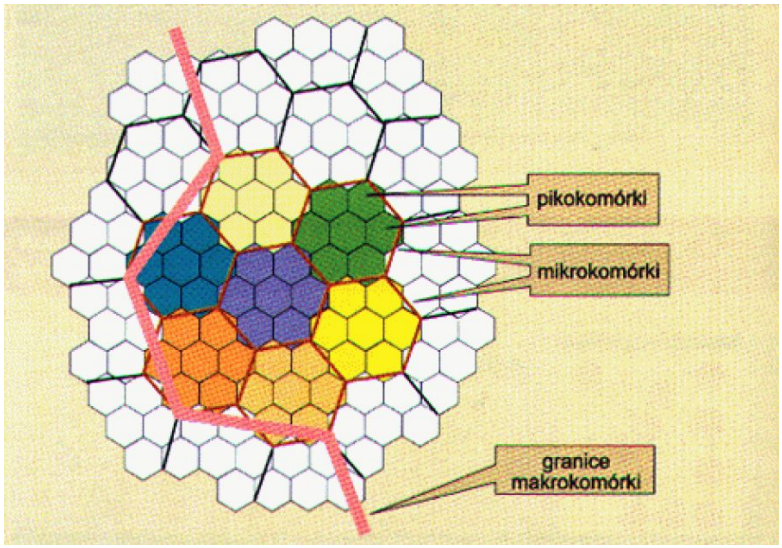
- komórkowa struktura sieci. Sieć składa się z szeregu urządzeń nadawczo-odbiorczych (tzw. komórek), z których każde jest obsługiwane przez określoną stację bazową. Komórki te są różnych rozmiarów, w zależności od stopnia skomplikowania terenu i skupienia abonentów,

- ciągła aktualizacja stanu aktywnych telefonów komórkowych znajdujących się w zasięgu określonej stacji bazowej. Aktualizacja ta ma na celu lokalizację przemieszczających się abonentów. Może być dokonywana automatycznie, na bieżąco — podczas inicjowania każdego połączenia, lub okresowo w trakcie przemieszczania się abonenta z zasięgu jednej stacji bazowej do drugiej.

Pierwszy wymieniony powyżej czynnik implikuje relacje między elementami takiej sieci, wśród których dwa są najważniejsze (w kontekście zapewnienia jej dostępności). Są to: stacja bazowa oraz jej kontroler. Istotnym jest przynajmniej pobieżne zrozumienie roli wymienionych elementów, ponieważ ich błędne funkcjonowanie prowadzi do zakłócenia lub nawet niedostępności usług (sygnału). W skład każdej stacji bazowej (ang. *base transceiver station*) wchodzą następujące elementy: komórki (ang. *cells*), wzmacniacz sygnału (ang. *power amplifier*), przełącznik antenowy (ang. *duplexer*), łącznik sygnału (ang. *combiner*), system kontrolno-alarmowy (ang. *alarm and control system*). Należy tutaj nadmienić, że stacja bazowa obsługuje wiele komórek oraz zapewnia bezprzewodową łączność między terminalem abonenta (telefonem) a infrastrukturą operatora.

Kolejnym istotnym urządzeniem, ściśle powiązanym ze stacjami bazowymi, jest ich kontroler (ang. *base station controller*), który odpowiada za logikę działania tych stacji. Do jego zadań należy m. in.: wybór i przydzielanie odpowiednich kanałów radiowych, kontrola przekazywania obsługi telefonu komórkowego z jednej stacji do drugiej, odbiór oraz przetwarzanie parametrów pomiarowo-identyfikacyjnych terminali klienckich [33]. Zwykle pod jednym kontrolerem pracuje od kilku do kilkuset stacji bazowych.

W celu zapewnienia optymalnego zasięgu na danym obszarze geograficznym, stacje bazowe powinny być ułożone na kształt plastra miodu. Cały obszar, który należy pokryć jest zatem dzielony na heksagonalne regiony, w centrum których znajdują się stacje bazowe [33]. Ilustruje to rysunek 2.1. Niestety w rzeczywistych warunkach, kształt ten jest daleki od idealnego. Charakter zabudowy i konfiguracji ulic, wysokość budynków, czy nieregularne ukształtowanie terenu ma znaczący (często negatywny) wpływ na zasięg oraz jakość połączenia. Przykładowo, w miastach i innych terenach zurbanizowanych odbiór sygnału jest utrudniony przez jego załamania oraz odbicia od płaszczyzn czy ścian budynków. W zależności od potrzeb takie obszary (charakteryzujące się również wysokim natężeniem ruchu telekomunikacyjnego) dzielone są umownie na mniejsze mikro czy pikokomórki, a system komórkowy wykorzystując procedury lokalizacyjne, ocenia mobilność każdego abonenta i przyporządkowuje go do właściwego, optymalnego obszaru komórkowego [13]. Dlatego też, poszczególne komórki muszą mieć anteny ustawione w różnych kierunkach i pod różnymi kątami. Implikuje to, że dwa urządzenia nadawczo-odbiorcze zlokalizowane blisko siebie pod kątem geograficznym, mogą charakteryzować się bardzo różnymi parametrami pracy. Zatem nie można wprost traktować położonych blisko siebie komórek jako podobne, nawet jeżeli będą pochodziły od jednego producenta i pra-



Rysunek 2.1: Tworzenie obszarów komórkowych.
Źródło: [13]

cowały pod tym samym kontrolerem (co nie było oczywiste przed poznaniem podstawowej wiedzy dziedzinowej). Po przedstawieniu podstawowych informacji na temat telefonii komórkowej, można przystąpić do opisu poszczególnych zbiorów danych badanych w rozprawie.

2.2 Struktura zestawów danych

Pierwszy zestaw danych *cell_loss*, stanowiący przedmiot analiz, agregował dane odnośnie urządzeń nadawczo-odbiorczych rozlokowanych w aglomeracji śląskiej, pochodzące z okresu od kwietnia do stycznia 2011 roku. Składa się ostatecznie z 143486 obiektów[1] zapisanych w jednej tabeli. Pomiar dostępności danej komórki dokonywany był w godzinnych interwałach czasu. Struktura każdego rekordu danych jest następująca[2]:

- *cellname* — identyfikator określonej komórki,
- *obszarId* — identyfikator obszaru geograficznego, w którym zlokalizowana jest dana komórka,
- *sektorId* — identyfikator sektora kierunku, w którym komórka nadaje,
- *kontrolerId* — identyfikator kontrolera, który steruje pracą danej komórki,
- *dostawcaId* — identyfikator producenta danej komórki,
- *wagaStraty* — procent niedostępności komórki (po dyskretyzacji[3]),

[1] Pojęcie obiektu jest w tym przypadku utożsamiane z pojedynczym rekordem tabeli.
[2] We wszystkich opisach tabel pominięto pole *Id* będące unikalnym identyfikatorem, które pozwala na rozróżnienie każdego rekordu, ponieważ nie jest ono istotne pod kątem prowadzonej analizy danych.
[3] Dyskretyzacja została wykonana przez eksperta dziedzinowego na pięć rozłącznych klas wartości.

- *strata*[4] — bezwzględny procent niedostępności danej komórki w danej godzinie,
- *zdarzenieId* — identyfikator zdarzenia[5],
- *start* – dzień i godzina zajścia określonego zdarzenia,
- *koniec* – dzień i godzina końca zdarzenia,
- *czasTrwaniaH* — czas trwania określonego zdarzenia wyrażony w godzinach,
- *czyPlanowane* — określa czy dane zdarzenie było zaplanowane,
- *data* – data (dzień, miesiąc, rok) dokonania pomiaru,
- *czestosc* — liczba powstałych (zaistniałych) zdarzeń w ciągu całego dnia, związanych z pracą danej komórki,
- *czyProblem* — określenie czy występuje jakiś problem z daną komórką[6],
- *typPrbId* — definiuje typ problemu jaki wystąpił z daną komórką[7],
- *czyWorkflow* — zostało wystawione zlecenie na dokonanie prac przy komórce,
- *czyWoINN* — dział utrzymania sieci ma zlecenie na wykonywanie prac przy danej komórce,
- *czyWoTeren* — inny dział poza utrzymaniem sieci ma zlecenie na wykonywanie prac przy danej komórce.

W tabeli danych występuje pięć atrybutów dychotomicznych (*czyProblem*, *czyWoINN*, *czyWoTeren*, *czyWorkflow*, *czyPlanowane*), które ze względu na swoją specyfikę utrudniają grupowanie (a konkretnie prawidłowe wyliczenie podobieństwa dwóch rekordów między sobą). Ponadto występuje również pięć atrybutów niezmiennych dla określonej komórki takich jak: jej identyfikator, identyfikator kontrolera, sektora, dostawcy i obszaru geograficznego na którym pracuje dana komórka. Przykładowy rekord omawianego zbioru danych *cell_loss* został zaprezentowany w postaci tabeli 2.1.

Pierwotnie opisywany zestaw danych miał nieco inną strukturę niż pokazano powyżej. W szczególności brak było atrybutów definiujących kierunek nadawania czy też przedział czasowy rejestrowanych zdarzeń. Po wykonaniu wstępnych eksperymentów z użyciem algorytmu DBSCAN (na zmniejszonej próbce danych, opisanych w [72]) zauważono, że niektóre wygenerowane grupy posiadają bardzo podobny opis deskryptorowy. Istotnie, jedynym rozróżniającym

[4] Atrybut ten nie jest uwzględniany podczas zadania grupowania, ponieważ posiada zbyt dużo unikalnych wartości. Zamiast tego podczas analizy skupień wykorzystywana jest *wagaStraty*. Jednakże *strata* została zachowana w strukturze tabeli, ponieważ po utworzeniu skupień analityk danych może chcieć uzyskać dokładnej informacji na temat tego parametru.

[5] Zdarzenie jest rejestrowane, gdy komórka poddawana jest naprawom bądź jest niedostępna z jakiegokolwiek powodu.

[6] Jest to wielkość uzupełniana przez operatora systemu monitorującego. Komórka mogła bowiem zostać wyłączona celowo ze względu na zaplanowane prace, bądź też z innych powodów.

[7] Wyróżnia się cztery typy problemów: dotyczący zasilania, transmisji, sprzętowy i inny.

Tabela 2.1: Przykładowy rekord ze zbioru danych *cell_loss*

Nazwa pola	Wartość
id	1
cellname	58294A1
obszarId	41
sektorId	582941
kontrolerId	145
dostawcaId	4
strata	0,0282142851501703
wagaStraty	1
zdarzenieId	646808
start	2010-06-23 04:00:00
koniec	2010-06-23 05:00:00
czasTrwaniaH	1
czyPlanowane	0
data	2010-06-23 00:00:00
czestosc	2
czyProblem	1
typPrbId	4
czyWorkflow	1
czyWoINN	1
czyWoTeren	0

czynnikiem obiektów należących do niektórych skupień były różne czasy rozpoczęcia pomiarów. Sformułowano zatem hipotezę, że istnieją zdarzenia dotyczące określonej komórki, które trwały dłużej niż godzinę, jednakże stan urządzenia nadawczo-odbiorczego (pod kątem rejestrowanych w zbiorze parametrów) nie ulegał zmianie. Hipoteza została potwierdzona przez eksperta z firmy telekomunikacyjnej udostępniającej dane. Poczyniona obserwacja miała dość duże znaczenie, ponieważ na jej podstawie możliwe było zredukowanie liczby danych, poprzez agregację podobnych rekordów do jednego, zapamiętując jednocześnie czas trwania zdarzenia. W konsekwencji zbiór danych został zmniejszony o ponad 16% w stosunku do pierwotnej postaci, bez utraty jakichkolwiek informacji.

Wprowadzenie do ostatecznej struktury drugiego atrybutu, powiązane było z kryterium doboru właściwej miary podobieństwa dwóch obiektów[8]. Obecność atrybutów jakościowych w zbiorze utrudniała ustalenie jakiejkolwiek relacji porządku. Przykładowo identyfikatory 5001A1 i 5002A1 różnią się tylko jedną pozycją (w sensie numerycznym), jednakże mogą odnosić się do dwóch bardzo różnych (w kontekście rozmieszczenia i parametrów) komórek. O skuteczności procesu analizy skupień często decyduje właściwy wybór miary podobieństwa (odległości) obiektów grupowanych. Dlatego też biorąc pod uwagę złożoność analizowanego zbioru danych *cell_loss* oraz fakt, że dane są wielotypowe zdecydowano się na zdefiniowanie miary podobieństwa jako liczby cech wspólnych grupowanych elementów. Jednocześnie autor

[8]Niniejsza praca zakłada, że obiekty będą poddawane analizie skupień, co zostanie umotywowane w dalszej części tego rozdziału.

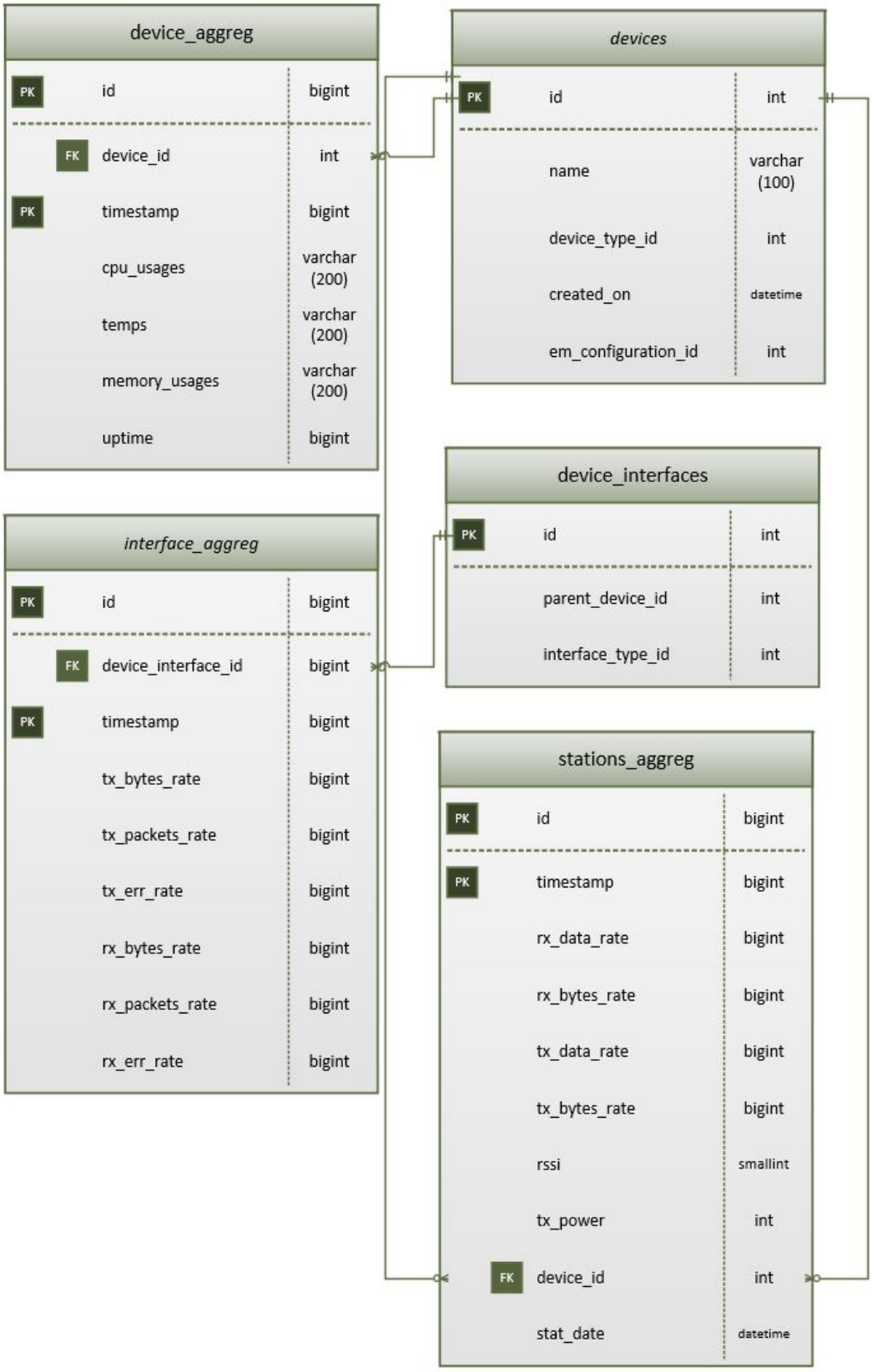
przystąpił do zbadania, które atrybuty powinny być brane pod uwagę w dalszym etapie analizy i procesie grupowania, ze szczególnym uwzględnieniem identyfikatorów. Niektóre z nich, jak informacje o kontrolerze czy regionie muszą być uwzględnione podczas grupowania, ponieważ potencjalnie komórki działające pod tym samym kontrolerem są ze sobą skorelowane, podobnie jak te umiejscowione w tym samym obszarze geograficznym. Zatem jedynymi atrybutami, które zostały uznane za potencjalnie niepotrzebne (do uwzględnienia podczas grupowania), były *cellname* oraz *zdarzenieId*. Jednakże by mieć pewność, że nieuwzględnienie tych atrybutów nie powoduje utraty jakichkolwiek informacji czy zależności, skontaktowano się z dostawcą zestawu danych. Odkryto, że w identyfikatorze komórki zakodowana jest również informacja dotycząca kierunku nadawania sygnału. Dlatego też uwzględniono informacje o sektorach w ostatecznej strukturze.

Kolejny pozyskany zbiór danych *ap_loss* pochodzi bezpośrednio od klientów jednej z firm produkujących oprogramowanie do zarządzania sieciami bezprzewodowymi i składa się z pięciu tabel (których schemat połączeń został zaprezentowany na rysunku 2.2), zawierających najistotniejsze informacje odnośnie parametrów transmisji czy użycia zasobów przez poszczególne urządzenia sieciowe. Wspomniane tabele pierwotnie opisane były za pomocą od 11 do 42 atrybutów oraz agregowały od 50 do 1697607 obiektów. W przeciwieństwie do pierwszego zbioru, występują tutaj wartości puste jak również nie ma kompletu danych o wszystkich urządzeniach zapisanych w bazie. Dlatego też do dalszych analiz wybrano jedynie te rekordy, które odnosiły się do znanych identyfikatorów urządzeń – wyeliminowano wpisy, które dotyczyły nieistniejących (usuniętych) węzłów sieci. Wspomniany krok pozwolił na ograniczenie (maksymalnie o około 43 procent) liczby obiektów dla omawianego zestawu danych. Ostatecznie tabele (omówione w dalszej części tej sekcji) wchodzące w skład zbioru *ap_loss* przechowują kolejno: 406799 (*interface_aggreg*), 961281 (*stations_aggreg*), 204258 (*device_aggreg*), 50 (*devices*) rekordów[9]. Jednakże ze względu na duży rozmiar danych, nie jest możliwe (biorąc pod uwagę ograniczenia sprzętowe[10]) złączenie wszystkich pięciu tabel, na podstawie odpowiednich kluczy, w rozsądnym czasie. Implikuje to konieczność pracy z mniejszą liczbą danych jednocześnie i ich analizę pod ściśle określonym kątem. Przykładowo, badając które urządzenia mają największe obciążenie procesora, nie są potrzebne informacje dotyczące parametrów ich transmisji. Problem ten oczywiście nie występuje w pierwszym zbiorze danych, ponieważ cały zestaw udało się przedstawić w formie tylko jednej tabeli.

Pierwsza tabela *devices* zawiera informacje niezmienne dla wszystkich przechowywanych w bazie urządzeń jak ich nazwa, typ, data dodania do systemu itp. Początkowo były one opisane za pomocą 42 atrybutów (bez uwzględnienia sztucznie wprowadzonych pól identyfikatorów rozróżniających rekordy w tabeli), jednakże duża część z nich została pominięta w dalszych analizach za sprawą selekcji dokonanej przez eksperta oraz występowania wartości pustych (których nie da się odtworzyć) bądź stałych. Przykładowo atrybut *opis urządzenia* jest opcjonalny, co w tym przypadku oznacza, że tylko jedno urządzenie ma to pole wypełnione, ponieważ

[9] Tabelę *device_interfaces* pominięto w analizach, ponieważ pełniła ona wyłącznie rolę encji asocjacyjnej. Szczegóły na ten temat zawarte są w dalszej części niniejszego rozdziału.

[10] Autor postanowił stworzyć oprogramowanie, które jest w stanie dokonać analizy danych złożonych wykorzystując, w możliwie największym zakresie, parametry sprzętowe (niemalże) każdego komputera. Dlatego nie będą wykorzystywane zasoby stacjonarnych centrów obliczeniowych bądź przetwarzanie w chmurze.



Rysunek 2.2: Uproszczony diagram ERD dla tabel wchodzących w skład zbioru *ap_loss*.
Źródło: Opracowanie własne

Tabela 2.2: Przykładowy rekord tabeli *devices* wchodzącej w skład zbioru *ap_loss*

Nazwa pola	Wartość
id	1
name	AP-Luxor-1
device_type_id	9
created_on	2009-07-04 10:03:41.000
em_configuration_id	17

tego typu dane są uzupełniane do systemu przez jego użytkowników zgodnie z ich uznaniem. Brakuje także danych lokalizacyjnych (w postaci długości, szerokości i wysokości geograficznej) oraz kilku innych. Występują również atrybuty jednowartościowe (jak chociażby numer portu do zarządzania), co jest niekorzystne z punktu widzenia analityka. Dodatkowo niektóre identyfikatory nie występują w żadnych innych tabelach opisywanego zestawu danych. Dlatego też zostały one usunięte, co zmniejszyło liczbę rekordów (urządzeń) do pięćdziesięciu. Ostateczna struktura opisywanej tabeli prezentuje się zatem następująco:

- *name* – nazwa urządzenia sieciowego[11],
- *device_type_id* – identyfikator określający typ urządzenia sieciowego. Możliwe jest przyporządkowanie do jednego z 45 typów, jednakże w bazie wykorzystywane są tylko trzy,
- *created_on* – dokładna data i czas dodania urządzenia sieciowego do systemu monitorującego,
- *em_configuration_id* – numer konfiguracji na jaką urządzenie ma się przełączyć w sytuacji zidentyfikowania przez czujniki awarii.

Tabela 2.3: Przykładowy rekord tabeli *device_aggreg* wchodzącej w skład zbioru *ap_loss*

Nazwa pola	Wartość
id	55358309342
device_id	1
timestamp	1285357954501
cpu_usages	42
temps	67
memory_usages	23
uptime	166531

Kolejna tabela o nazwie *device_aggreg* zawiera informacje na temat wykorzystania zasobów (czasu procesora czy zużycia pamięci) przez wszystkie urządzenia opisane za pomocą 11 atrybutów. Do analiz (z podobnych względów jak w poprzedniej tabeli) zostały wybrane jednak następujące:

- *device_id* – unikalny identyfikator urządzenia w obrębie bazy,

[11] Nazwa nie musi być unikalna i jest nadawana przez użytkownika.

- *timestamp* – data i czas dokonania pomiaru wyrażona w postaci tzw. czasu uniksowego (ang. *unix timestamp*),
- *cpu_usages* – wykorzystanie procesora w skali od zera do stu procent,
- *temps* – aktualna temperatura urządzenia wyrażona w Celsiuszach,
- *memory_usages* – zużycie pamięci w skali od zera do stu procent,
- *uptime* – czas nieprzerwanej pracy urządzenia wyrażony w sekundach.

Tabela 2.4: Przykładowy rekord tabeli *interface_aggreg* wchodzącej w skład zbioru *ap_loss*

Nazwa pola	Wartość
id	55358306888
device_interface_id	1
timestamp	1285357954506
tx_bytes_rate	392
tx_packets_rate	1118
tx_err_rate	0
rx_bytes_rate	523
rx_packets_rate	4984
rx_err_rate	0

Trzecia i czwarta tabela, tj. *device_interfaces* oraz *interface_aggreg* są ze sobą silnie skorelowane. Mianowicie *device_interfaces* przechowuje dane odnośnie nazw czy przypisanych adresów ip dla poszczególnych interfejsów sieciowych (wejścia-wyjścia) każdego z urządzeń, natomiast *interface_aggreg* agreguje statystyki związane z ruchem na danym interfejsie (jak liczba wysłanych pakietów czy błędów transmisji). Tabela *device_interfaces* będzie wykorzystywana wyłącznie jako pomost między *devices* oraz *interface_aggreg*, ponieważ urządzenia mają różna liczbę interfejsów (niektóre tylko jeden port, inne po dwa), a z punktu widzenia analityka danych ważne jest postrzeganie danego urządzenia jako całości. Nie znana jest też struktura połączeń ani konfiguracja całej sieci. Dlatego po wykryciu urządzeń, których praca odstaje od reszty, dopiero ekspert dziedzinowy może dokonać dalszej oceny (badając np. ruch na poszczególnych złączach). Aby ułatwić analizę omawianego zestawu danych za pomocą algorytmów analizy skupień, dodano do schematu tabeli *interface_aggreg* dodatkową kolumnę *device_id* (powstałą na skutek skorelowania informacji zawartych w *device_interfaces* korzystając z kluczy obcych i głównego), która określa do jakiego urządzenia należy dany interfejs. Dzięki temu nie jest konieczne wykorzystanie encji asocjacyjnej *device_interfaces* w dalszych analizach. Struktura tabeli *interface_aggreg* po usunięciu atrybutów jednowartościowych oraz nieistotnych z punktu widzenia analizy danych prezentuje się następująco:

- *device_interface_id* – unikalny identyfikator danego interfejsu,
- *timestamp* – data i czas dokonania pomiaru wyrażona w postaci czasu uniksowego,
- *tx_bytes_rate* – liczba wysyłanych danych mierzona w bajtach,

- *tx_packets_rate* – liczba transmitowanych pakietów,
- *tx_err_rate* – zarejestrowana liczba błędów transmisji,
- *rx_bytes_rate* – liczba odbieranych danych wyrażona w bajtach,
- *rx_packets_rate* – liczba odbieranych pakietów,
- *rx_err_rate* – zarejestrowana liczba błędów odbioru,
- *device_id* – identyfikator urządzenia, do którego należy dany interfejs.

Szczególna uwaga zostanie poświęcona atrybutom *tx_err_rate* oraz *rx_err_rate*, na podstawie których będzie można potencjalnie oszacować urządzenia odstające od normy pod kątem liczby błędów odbioru i transmisji. Dzięki temu, ekspert będzie mógł zweryfikować prawidłowość konfiguracji sieci, warunki pracy tych urządzeń bądź inne czynniki wpływające na ich awaryjność i występowanie błędów.

Tabela 2.5: Przykładowy rekord tabeli *stations_aggreg* wchodzącej w skład zbioru *ap_loss*

Nazwa pola	Wartość
id	62186668074
timestamp	1285357954514
rx_data_rate	0
rx_bytes_rate	0
tx_data_rate	0
tx_bytes_rate	0
rssi	0
tx_power	0
device_id	2
stat_date	2010-09-24 19:58:22.000

Ostatnia tabela *stations_aggreg* agreguje dane klientów (np. urządzeń typu punkt dostępowy) podłączonych do stacji bazowej. Znajdują się w niej informacje przede wszystkim o liczbie wysyłanych i odbieranych danych, jak również poziomie sygnału. Liczba przechowywanych obiektów (rekordów) jest największa spośród omówionych zestawów i wynosi ponad półtorej miliona, co utrudnia jakiekolwiek przetwarzanie, ponieważ samo wyświetlenie danych (korzystając z przystosowanego do tego oprogramowania bazodanowego) wynosi około 22 sekund[12]. Przyjęty schemat tabeli jest następujący:

- *timestamp* – data i czas dokonania pomiaru wyrażona w postaci czasu uniksowego,
- *rx_data_rate* – liczba odebranych przez stację bazową pakietów,
- *rx_bytes_rate* – liczba odebranych bajtów przez stację bazową,

[12]Pomiar dokonany był na komputerze wyposażonym w 2,5GHz procesor Core i5-3210M oraz 6GB pamięci RAM.

- *tx_data_rate* – liczba wysłanych pakietów przez stację bazową,
- *tx_bytes_rate* – liczba bajtów transmitowanych przez stację,
- *rssi* – wskaźnik mocy odbieranego sygnału radiowego mierzona w dBm[13],
- *tx_power* – siła sygnału nadawania mierzona w dBm,
- *device_id* – identyfikator urządzenia podłączonego do stacji,
- *stat_date* – data i czas zapisania pomiaru z dokładnością do jednej sekundy.

Biorąc pod uwagę różnice i specyfikę przedstawionych rzeczywistych zbiorów można stwierdzić, że wydobywanie wiedzy z danych rozpoczyna się już na etapie ich przygotowania do analizy. W literaturze przedmiotu [7, 66, 28] wyróżnia się następujące techniki analizy danych: *znajdowanie asocjacji*, *selekcję istotnych atrybutów*, *dyskretyzację*, *wykrywanie odchyleń* oraz *grupowanie* danych. Wyszukiwanie asocjacji w danych pozwala odnaleźć powiązania między cechami (co jest przydatne podczas selekcji potencjalnie istotnych atrybutów) na podstawie ich współwystępowania w zbiorze treningowym. Najczęściej powiązania te wizualizowane są za pomocą reguł asocjacyjnych [22]. Ekstrakcja i selekcja cech polega na wyborze jedynie pewnego minimalnego podzbioru cech, tak by usunąć cechy redundantne bądź nieistotne z punktu widzenia realizowanego zadania eksploracji danych. Pozwala to na znaczne ograniczenie zajmowanej pamięci oraz potencjalnie poprawia jakość rezultatu wygenerowanego przez algorytm eksploracji danych. Dyskretyzacja i kodowanie danych opiera się na podziale zbioru wartości atrybutu na zestaw ściśle określonych klas. Zastosowanie dyskretyzacji bądź zakodowanie danych do innego formatu zapisu umożliwia łatwiejszą interpretację badanej struktury, a przede wszystkim redukcję zbioru. Wykrywanie braków i odchyleń w danych polega na wykryciu wartości błędnych, brakujących bądź wyraźnie odstających na tle wszystkich pozostałych. Brakujące wartości można uzupełnić wartością najczęściej występującą bądź uśrednioną. Można także zastosować strategię lokalną (np. stosując algorytm k-najbliższych sąsiadów) i badać jedynie wartości będące w najbliższym sąsiedztwie względem analizowanej[14]. Segmentacja danych natomiast opiera się na podziale zbioru źródłowego na grupy, poprzez zastosowanie odpowiedniego algorytmu analizy skupień. Dla każdego skupienia tworzeni są reprezentanci, którzy mogą zostać wykorzystani do opisu struktury danych bądź jako forma ich kompresji — w dalszych etapach analizy tylko reprezentanci grup będą wówczas brani pod uwagę.

W niniejszej rozprawie zostaną wykorzystane niektóre z przedstawionych technik, a mianowicie analiza częstości występowania danej cechy, segmentacja danych, czy wykrywanie anomalii, ponieważ to właśnie te metody potencjalnie pozwolą wydobyć najwięcej wiedzy oraz informacji o strukturze danych wejściowych. Stanowi to bezpośrednie nawiązanie do metod opisu danych (analizowanych szczegółowo w rozdziale 4) oraz procesu graficznej analizy eksploracyjnej (przedstawionego w rozdziale 6). Ponadto, pewne czynności jak wybór cech istotnych, powinny być wykonywane przez ekspertów (dostawców danych), ze względu na ich doświadczenie i wiedzę dziedzinową, która jest niezbędna podczas analizy rzeczywistych zbiorów danych, by wysunąć prawidłowe wnioski z badań oraz prawidłowo określić ich główny cel.

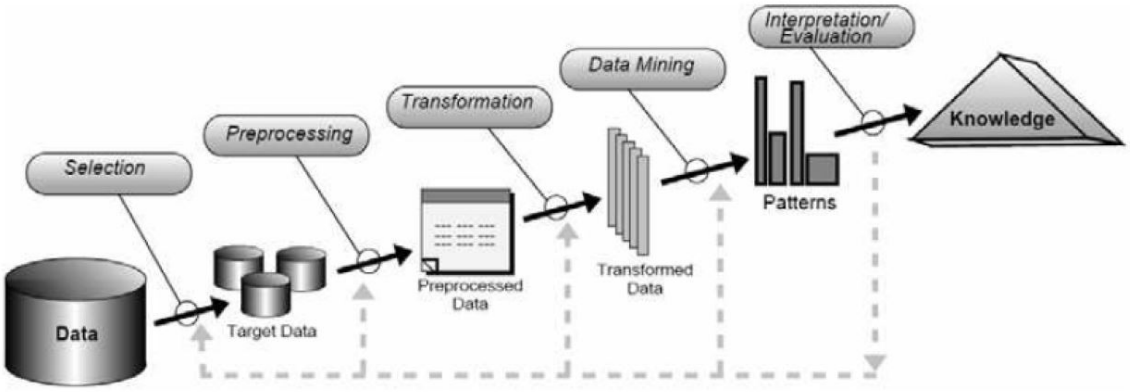
[13] Jednostka dBm to (logarytmiczna) miara mocy odniesiona do jednego miliwata.
[14] W literaturze [66] można znaleźć wiele metod wykrywania odchyleń w danych.

2.3 Iteracyjny proces odkrywania wiedzy

Przetwarzanie rzeczywistych zbiorów danych złożonych powinno mieć charakter iteracyjny – po zastosowaniu konkretnej techniki eksploracji danych i wizualizacji rezultatów, analiza wyników prócz ich oceny bardzo często może sugerować zmianę celu badawczego bądź modyfikację struktury danych (tak jak to miało miejsce przy redukcji zbioru *cell_loss* odnośnie telefonii komórkowej). Dlatego też w dalszej części rozdziału zostanie omówiony proces odkrywania wiedzy KDD (ang. ***K**nowledge **D**iscovery in **D**atabases*) wraz z odniesieniem się do planowanego rozwiązania eksploracji danych złożonych.

Klasyczny proces odkrywania wiedzy (przedstawiony na rysunku 2.3) składa się z pięciu następujących etapów [1]:

1. zapoznanie się z dziedziną i specyfiką problemu, który będzie poddawany analizie oraz ustalenie celu odkrywania wiedzy,
2. utworzenie docelowego zbioru danych poprzez selekcję istotnych informacji,
3. czyszczenie i przetwarzanie danych,
4. wybór i zastosowanie właściwego zadania oraz algorytmu eksploracji danych,
5. analiza i interpretacja wygenerowanych wzorców oraz ich zastosowanie.



Rysunek 2.3: Etapy klasycznego procesu odkrywania wiedzy.
Źródło: [1]

Pierwszym krokiem procesu odkrywania wiedzy jest szczegółowe zapoznanie się ze specyfiką analizowanego problemu, jak i uświadomienie sobie jaki cel chcemy osiągnąć używając narzędzi eksploracji danych. Ważnym jest także zrozumienie wszelkich wymagań określanych z biznesowego punktu widzenia oraz ich odpowiednie przekształcenie na problem zrozumiały dla analityka danych. Niejednokrotnie może się to okazać zadaniem skomplikowanym, ze względu na to, że w pierwszym przypadku posługujemy się językiem stricte biznesowym, często

specyficznym dla danej instytucji, podczas gdy analitycy danych używają wyłącznie terminologii naukowo-technicznej. Przykładowym celem biznesowym może być *zwiększenie sprzedaży określonych towarów oferowanych przez firmę*. Odpowiednikiem tego celu, z punktu widzenia zadania eksploracji danych, jest *prognoza liczby kupowanych artykułów, przy wykorzystaniu analizy koszykowej zakupów klientów oraz powiązanych informacji statystycznych i demograficznych (jak płeć, miesięczny dochód itp.)*. Bardzo ważne jest zatem właściwe przekonwertowanie celów i warunków biznesowych, na odpowiadające im sformułowania wyrażone w języku zrozumiałym dla analityków danych, gdyż ma to ogromny wpływ na ostateczny rezultat omawianego procesu odkrywania wiedzy, a w konsekwencji na poprawność znalezionych zależności i trendów. Rola wiedzy dziedzinowej została zaakcentowana również w poprzednich sekcjach tego rozdziału.

Faza druga procesu odkrywania wiedzy rozpoczyna się od zgromadzenia wszelkich relewantnych danych, zapisanych w różnych postaciach oraz wstępnego zapoznania się z ich strukturą, w celu zidentyfikowania nieprawidłowości w zestawie danych (takich jak braki pewnych wartości czy błędy). Ponadto następuje wyselekcjonowanie interesujących podzbiorów danych oraz ich wstępne przetworzenie, które obejmuje transformację danych (np. do jednolitego, określonego formatu) oraz ich czyszczenie (usuwanie wartości błędnych, pustych).

Krok trzeci składa się z wyboru atrybutów znaczących z punktu widzenia zadania eksploracji danych. Co więcej uwzględnia on techniki redukcji liczby wymiarów danych, w celu późniejszej, łatwiejszej interpretacji ich natury i powiązań między nimi.

Kolejny etap to wybór odpowiedniego zadania eksploracji danych oraz odpowiednio przystosowanego algorytmu do wydobywania wzorców, trendów czy relacji dla danego zbioru. Może być on realizowany w sposób nadzorowany bądź nie, w zależności od wymagań oraz wybranej metody.

Następny krok obejmuje analizę i interpretację otrzymanej wiedzy w postaci wzorców czy relacji. Oceniana jest jej poprawność jak również potencjalna przydatność. Ponadto może zostać poddana wizualizacji, w celu łatwiejszego zrozumienia i wyjaśnienia występujących zależności (co ma bardzo często miejsce w kontekście analiz danych rzeczywistych). Należy tutaj zwrócić uwagę na fakt, że owa ocena jest dokonywana dwustopniowo. Najpierw analityk danych sprawdza zgodność oraz jakość uzyskanych wzorców, na podstawie własnej wiedzy dziedzinowej, metod oceny charakterystycznych dla danej metody eksploracji danych oraz zdefiniowanych początkowo wymagań. Następnie kontaktuje się on z analitykiem biznesowym lub ekspertami z danej dziedziny, w celu dyskusji nad otrzymanymi rezultatami w kontekście przydatności biznesowej i realizacji założonych celów.

Przedstawiony proces ma charakter iteracyjny, co jak zaznaczono na rysunku 2.3 symbolizuje strzałka w dół biegnąca od kroku interpretacji i ewaluacji wyników do dowolnego pozostałego etapu. Jednakże autor proponuje tą koncepcję rozszerzyć: powrót powinien być możliwy z każdej fazy procesu wydobywania wiedzy do dowolnej innej. Ponadto faza transformacji i przetwarzania powinna zostać rozszerzona o elementy metod opisu danych (zarówno wykorzystujących statystykę opisową jak i techniki graficzne), dzięki czemu wydobywanie wiedzy może odnieść skutek już na etapie zapoznawania się z danymi. Przykładowo w kontekście zbioru urządzeń sieciowych można powiązać liczbę błędów transmisji z konkretnymi urządze-

niami, a następnie dokonując wizualizacji na histogramie, zobaczyć czy któreś z nich nie odstaje od reszty (pod kątem tego parametru). Zatem jeszcze przed realizacją właściwego zadania eksploracji (jak np. analizy skupień) można zidentyfikować obiekty szczególnego zainteresowania. Ponadto zastosowanie statystyki opisowej i wyznaczenie liczby unikalnych wartości dla wszystkich atrybutów jest pomocne przy wyborze tych, które będą brały udział w dalszej analizie – na tej podstawie usuwa się wartości puste oraz atrybuty jednowartościowe. Podobna zasada była stosowana do obu analizowanych w niniejszej pracy zestawów danych. Koncepcja wydobywania wiedzy z danych złożonych proponowana przez autora pracy składa się zatem z następujących etapów:

1. selekcja cech na podstawie oceny eksperckiej,
2. identyfikacja rozkładu przyjmowanych wartości dla analizowanych cech, celem rozpoznania atrybutów dychotomicznych, wartości brakujących, zduplikowanych, lub odstających,
3. dyskretyzacja wartości atrybutów ilościowych przy konsultacji z ekspertem,
4. zastosowanie gęstościowego algorytmu analizy skupień, celem wygenerowania reprezentantów utworzonych grup,
5. wizualizacja wyników analizy skupień za pomocą techniki map prostokątów,
6. wykorzystanie utworzonych reprezentantów jako uogólnienia wzorców do dalszego procesu ekstrakcji wiedzy.

Pierwszym etapem prac podczas analizy dużych, rzeczywistych zbiorów jest selekcja cech istotnych z punktu widzenia rozważanego problemu, dokonywana przez eksperta. Ze względu na złożoną strukturę takich zbiorów oraz konieczność posiadania często rozległej wiedzy dziedzinowej, udział eksperta w tym procesie jest rzeczą niezbędną.

Następnie rozpoczyna się etap przygotowania danych do analizy uwzględniający m.in. wykrycie wartości pustych, zduplikowanych, odstających czy ogólnego rozkładu dla danej cechy. W przeciwieństwie jednak do schematu wykorzystywanego w literaturze, etap ten ma pozwalać również na wykrycie wiedzy czy zależności z danych, przy wykorzystaniu metod opisu danych (omówionych w rozdziale 4). Dalsza część proponowanego rozwiązania polega na dokonaniu dyskretyzacji atrybutów ilościowych, z uwzględnieniem zdania eksperta o możliwości i zasadności dokonania takiego podziału.

Kolejnym krokiem jest zastosowanie algorytmu gęstościowego celem wygenerowania struktury skupień i ich reprezentantów. Struktura ta powinna zostać przedstawiona użytkownikowi w formie czytelnej i łatwiej do interpretacji. Zostanie to osiągnięte między innymi przez wykorzystanie techniki map prostokątów (przedstawionej w rozdziale 6). Wygenerowani reprezentanci stanowią uogólnienie wiedzy zawartej w danych, dzięki czemu możliwe jest zastosowanie innej techniki ekstrakcji wiedzy (co nie było wcześniej możliwe ze względu na rozmiar danych) na ograniczonym zbiorze wejściowym – wyłącznie do reprezentantów skupień.

2.4 Podsumowanie

Celem rozdziału drugiego było przedstawienie problemów występujących przy zagadnieniu wydobywania wiedzy z danych złożonych, na rzeczywistych przykładach zbiorów dotyczących telefonii komórkowej czy skomplikowanej sieci urządzeń bezprzewodowych. Szczególnym przedmiotem analizy była rola wiedzy dziedzinowej nie tylko w procesie rewizji, interpretacji i oceny uzyskanych ostatecznie rezultatów, lecz również jako czynnik, który ma znaczący wpływ na przebieg samej analizy eksploracyjnej. Zdefiniowano również podstawowe pojęcia wykorzystywane w telefonii komórkowej, co jest niezbędne do zrozumienia struktury i zależności występujących w zbiorach danych będących przedmiotem analizy badawczej. Przedstawiono także klasyczny proces odkrywania wiedzy wraz z odniesieniem się do planowanej metody eksploracji danych złożonych.

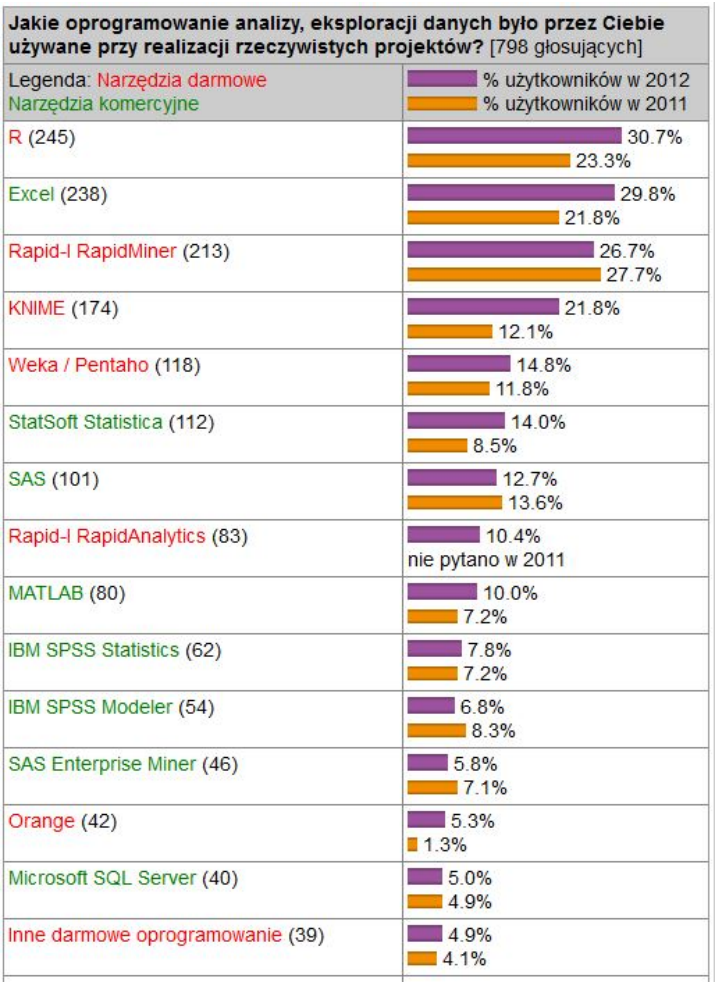
Rozdział 3

Analiza dostępnych rozwiązań programowych

Do skutecznego przeprowadzenia procesu odkrywania wiedzy, prócz dobrej znajomości badanej dziedziny, zbioru danych czy wybrania właściwej techniki eksploracji, niezbędnym jest wykorzystanie odpowiedniego oprogramowania. Od momentu rozpowszechnienia możliwości jakie dają komputery w połączeniu z właściwymi algorytmami, powstało wiele narzędzi wspomagających wydobywanie wzorców, trendów i relacji (jak na przykład *Statistica* firmy StatSoft, czy *Microsoft Analysis Services*), jednakże każde z nich posiada pewne cechy charakterystyczne odróżniające od pozostałych. Celem niniejszego rozdziału jest analiza porównawcza dostępnych rozwiązań programowych w zakresie odkrywania wiedzy, ze szczególnym uwzględnieniem oferowanych możliwości przetwarzania dużych zbiorów danych złożonych. Dokonany zostanie przegląd wybranych programów komercyjnych jak również tych, umożliwiających bezpłatne korzystanie z oferowanych możliwości. Na szczególną uwagę zasługuje tzw. otwarte oprogramowanie (ang. *open source software*), czyli z możliwością przestudiowania oraz dokonywania zmian w kodzie źródłowym. Dzięki idei otwartego oprogramowania, nie trzeba tworzyć bowiem całych systemów od początku, a jedynie dokonać stosownych zmian w zależności od indywidualnych potrzeb. Ponadto jeżeli dany projekt rozwijany jest przez dużą grupę programistów, a każdy zainteresowany ma wgląd w kod źródłowy, daje to większą szansę, że takie programy są w mniejszym stopniu narażone na błędy niż te tworzone od podstaw przez pojedyncze osoby. Niestety jakość kodu[1] tworzonego przez (często międzynarodową) społeczność jest bardzo zróżnicowana (co rzecz jasna ma spory wpływ podczas przetwarzania dużych zbiorów danych złożonych). Ponadto, wykorzystanie nawet relatywnie niewielkiego fragmentu cudzego kodu, wymusza najczęściej przywiązanie do konkretnego języka programowania (który wcale może nie być najlepszy z punktu widzenia stawianych wymagań) oraz rodzi konieczność zaimplementowania określonych struktur danych. Najczęściej im oprogramowanie oferuje większą

[1]Poprzez jakość kodu rozumie się wprowadzone optymalizacje oraz wykorzystane struktury danych celem zminimalizowania złożoności obliczeniowej lub zajętości pamięci danego algorytmu.

funkcjonalność, tym więcej zróżnicowanych kontenerów i struktur danych wykorzystuje, które trzeba dostosować do aktualnego stanu własnego programu. Dlatego też w podsumowaniu niniejszego rozdziału znajduje się motywacja do stworzenia autorskiego oprogramowania do wydobywania wiedzy z rzeczywistych danych złożonych.



Rysunek 3.1: Wyniki ankiety odnośnie używanego oprogramowania do analizy danych. Źródło: [76]

Wybór oprogramowania do analizy przeprowadzono na podstawie ankiety serwisu KDnuggets [76], agregującego artykuły, opinie i materiały edukacyjne odnośnie teorii oraz oprogramowania analizy danych. W maju 2012 roku zebrano odpowiedzi na pytanie jakiego narzędzia eksploracji i analizy danych użyłeś/aś w ciągu minionych 12 miesięcy podczas prac nad rzeczywistym projektem. Wyniki ankiety (wśród 798 głosujących) przedstawiono na rysunku 3.1 wraz z ich porównaniem do rezultatów otrzymanych rok wcześniej[2].

[2]Przedstawione wyniki zostały obcięte do pierwszych 15 pozycji ze względu na czytelność rysunku. Pełna wersja ankiety w języku angielskim dostępna jest pod adresem [76]. Należy również nadmienić, że możliwy był wybór kilku rozwiązań przez pojedynczego respondenta, dlatego wyniki nie sumują się do stu procent.

Pakiet R jest według badań serwisu KDnuggets najczęściej wybieranym systemem analizy danych (ponad 30% użytkowników potwierdziło korzystanie z niego podczas realizacji rzeczywistych projektów). Jest to zapewne zasługa bezpłatnej dystrybucji i sięgającej ponad 4000 liczbie dodatkowych bibliotek. Co zaskakujące, Excel jest niemalże równie popularny. W pierwszej chwili może się to wydawać nieuzasadnione biorąc pod uwagę możliwości konkurencyjnego oprogramowania nastawionego wyłącznie na zadania eksploracyjno-analityczne, jednakże należy pamiętać, że jest to narzędzie wchodzące w skład nawet najuboższych wersji pakietu MS Office, który jest bardzo często instalowany domyślnie na wielu komputerach. Zatem duże grono użytkowników posiada oraz potrafi bezproblemowo korzystać z funkcjonalności Excela. Badając wyniki ankiety, należy również nadmienić, że wśród pięciu najczęściej wybieranych programów, cztery to narzędzia otwarto-źródłowe, czyli udostępniające możliwość ingerencji w zasadę ich działania (kod źródłowy). Wśród narzędzi komercyjnych, obecnie największą popularność posiada pakiet Statistica firmy StatSoft, wyprzedzając tym samym system SAS amerykańskiego potentata SAS Institute. Jak stwierdzono w [76] jest to częściowo zasługa aktywnej kampanii marketingowej przeprowadzonej na zlecenie StatSoft. Niestety w przedstawionej ankiecie nie uwzględniono autorskich rozwiązań użytkowników, jednakże przeprowadzono również drugie badanie odnośnie języków programowania wykorzystywanych przy implementowaniu własnych narzędzi analizy danych. Rezultaty zilustrowano na rysunku 3.2.

Rysunek 3.2: Wyniki ankiety odnośnie używanego języka programowania do implementacji autorskich narzędzi analizy danych.

Źródło: [76]

Jak wynika z rysunku 3.2 ponownie największą popularnością cieszy się język programowania (i pakiet analizy danych) R. Z całą pewnością duży udział w tym sukcesie ma dobra dokumentacja, mnogość dodatkowych bibliotek oraz fakt, iż jest to pakiet stworzony od początku z myślą o statystyce oraz wizualizacji. Warto jednak nadmienić, że drugie miejsce zajmuje język operowania na relacyjnych bazach danych SQL. Potencjalnie jest to spowodowane faktem, że dane poddawane analizie są najczęściej przechowywane w tego typu bazach, przez co łatwiej jest na nich operować i generować statystyki wykorzystując bezpośrednio język SQL, niż tworzyć oprogramowanie, które połączy się z określoną bazą danych i pobierze odpowiednią porcję informacji do pamięci RAM oraz dokona jej przetworzenia czy analizy – może wprowa-

dzać to również pewne opóźnienia przy częstej komunikacji z serwerem bazodanowym. Podział na pozostałe języki programowania może wynikać z osobistych preferencji respondentów, dostępności gotowych rozwiązań w postaci zintegrowanych bądź dołączanych bibliotek, mnogości istniejących przykładów i literatury przedmiotu ukierunkowanej na analizę danych.

Biorąc pod uwagę przedstawione wnioski, do dalszej analizy zostanie wybrane oprogramowanie zajmujące czołowe miejsce ankiety zilustrowanej na rysunku 3.2 z którym autor niniejszej pracy miał styczność lub bezpośredni dostęp.

3.1 Oprogramowanie komercyjne

Wśród oprogramowania komercyjnego zostaną zbadane produkty firm IBM, Microsoft, StatSoft i SAS Institute. Podejście zastosowane w Microsoft Analysis Services wyróżnia się na tle pozostałych wysoką integracją z systemem bazodanowym SQL Server (którego jest częścią), dzięki czemu potencjalnie przyspiesza cały proces statystyczno-analityczny (wykorzystując indeksy i inne zoptymalizowane struktury do przechowywania danych jak również wyspecjalizowane mechanizmy przyspieszające dostęp do przechowywanych informacji), co zostanie omówione w odpowiedniej sekcji.

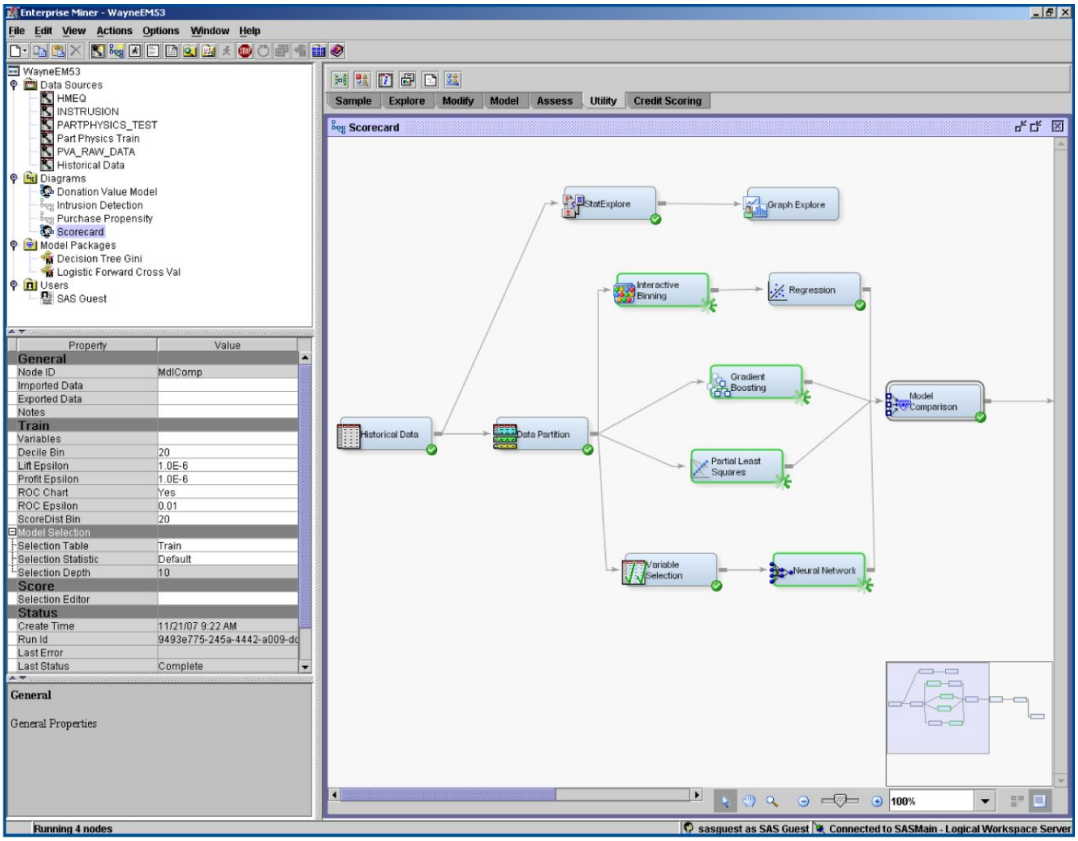
SAS Enterprise Miner

SAS Enterprise Miner jest produktem firmy SAS Institute i implementuje podejście SEMMA tj. proces próbkowania (ang. ***S**ampling*), eksploracji (ang. ***E**xploring*), modyfikacji (ang. ***M**odifying*), modelowania (ang. ***M**odeling*) oraz oceny (ang. **A**ssessing) dużych zbiorów danych. Wykorzystuje interfejs typu przeciągnij i upuść (ang. *drag and drop*) polegający na umieszczaniu połączonych ze sobą ikon, symulujących poszczególne kroki tworzenia modelu eksploracji i analizy danych [77]. Wygląd głównego okna omawianego programu prezentuje rysunek 3.3. Zgodnie z oficjalną dokumentacją i wymogami systemowymi[3] odnośnie instalacji SAS jest to pakiet liczący niespełna 40 różnych modułów, z których część może pracować autonomicznie (w tym wyszczególniony Enterprise Miner jako narzędzie do eksploracji danych). Mimo, że firma SAS Institute została utworzona w 1976 to jednak początki omawianego oprogramowania sięgają roku 1966 kiedy to późniejszy współzałożyciel Anthony J. Barr stworzył podstawową strukturę i język wykorzystywany do tworzenia SAS. Obecnie najnowsza wersja całej platformy to 9.4 (lub 12.3 w przypadku modułu Enterprise Miner) udostępniona w czerwcu 2013[4].

W broszurze opisującej możliwości produktu [77] można przeczytać, że "SAS Enterprise Miner udostępnia kompletny zbiór zaawansowanych narzędzi i algorytmów do modelowania predykcyjnego oraz opisowego. Wśród algorytmów analitycznych możemy wymienić: drzewa decyzyjne, sieci neuronowe, algorytmy bagging and boosting, MBR, regresję liniową i logistyczną, segmentację hierarchiczną, analizę asocjacji i sekwencji, analizę aktywności na stronach internetowych i wiele innych". Oprogramowanie wspiera użytkownika w całym procesie

[3] Wymagania systemowe odnośnie instalacji systemu SAS dostępne są pod adresem http://support.sas.com/documentation/installcenter/en/ikfdtnwx6sr/66390/PDF/default/sreq.pdf.

[4] Niestety autor niniejszej pracy nie miał bezpośredniego dostępu do najnowszej wersji systemu SAS i Enteprise Miner, dlatego zawarte w tej sekcji informacje oparto w dużej mierze o dostępne publicznie materiały i dokumentację producenta zawarte na stronie http://http://support.sas.com.



Rysunek 3.3: Interfejs programu SAS Enterprise Miner.
Źródło: [77]

odkrywania wiedzy począwszy od etapu przygotowania i transformacji danych (poprzez uzupełnianie wartości brakujących, wykrywanie odchyleń, wstępną segmentację zbioru) poprzez tworzenie i eksplorację modelu (generując np. szereg drzew decyzyjnych), aż do oceny uzyskanych rezultatów (wykorzystując odpowiednie wykresy i diagramy np. krzywych ROC). Eksploracja dużych zbiorów danych odbywa się poprzez zastosowanie metodologii przetwarzania rozproszonego (tam, gdzie to możliwe) dającej możliwość podziału najbardziej intensywnych obliczeniowo zadań na szereg wydzielonych maszyn serwerowych. Dzięki temu komputer kliencki nie jest niepotrzebnie obciążany, a wyniki uzyskiwane są wielokrotnie szybciej wykorzystując możliwości nowoczesnych procesów wielordzeniowych.

SAS Enterprise Miner implementuje bezpośrednio metody opisu danych[5], zarówno w formie statystyk typu mediana, moda, rozstęp międzykwartylowy jak również technik graficznych w postaci histogramów czy wykresów pudełkowych, pozwalające na zapoznanie się z analizowanym zbiorem na wczesnym etapie procesu wydobywania wiedzy. Ponadto wygenerowane wykresy najczęściej są interaktywne co oznacza, że użytkownik może dowolnie manipulować ich parametrami, jak również badać jaki wpływ na wygenerowane wizualizacje mają określone obiekty i parametry ze zbioru danych. Przykładowo badając zbiór zawierający informacje

[5]Metody opisu danych przedstawiono w rozdziale 4.

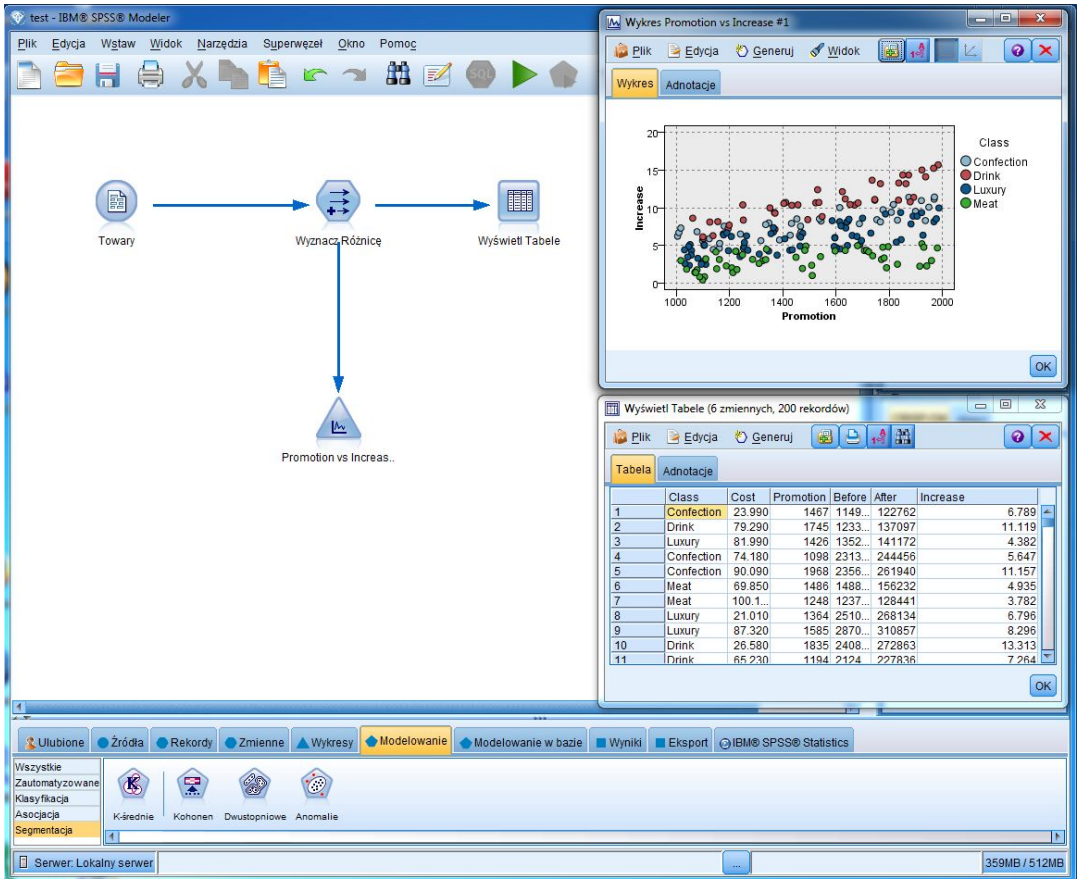
o sprzedawanych winach, użytkownik analizując histogram przedstawiający jakość wina (w skali od jeden do dziesięciu) może dowiedzieć się jak przedstawiają się wyniki jakości w podziale na wino białe i czerwone. Jest to analogia wykorzystywania zdolności kognitywnych człowieka w procesie graficznej eksploracji danych[6].

Z punktu widzenia problematyki przetwarzania danych złożonych, omawianej w niniejszej pracy, najsłabszą stroną programu SAS Enterprise Miner jest niewielka liczba zaimplementowanych algorytmów analizy skupień. Zgodnie z dokumentacją produktu [85] uwzględnione algorytmy pozwalają na otrzymanie hierarchicznej bądź płaskiej struktury grup (w tym wykorzystując niepełną przynależność obiektów do danego skupienia), jednakże tylko wartości numeryczne mogą być bezpośrednio przetwarzane przez procedury grupujące. Zostały stworzone jedynie funkcje wyznaczające macierz odległości dla danych numerycznych i tekstowych. Ponadto zaimplementowano jedynie cztery metody generowania skupień nazwane odpowiednio: *CLUSTER* (generującą strukturę hierarchiczną), *FASTCLUS* (opartą o algorytm niehierarchiczny k-średnich), *MODECLUS* (wykorzystującą nieparametryczną estymację gęstości), *VARCLUS* (uwzględniającą metody redukcji wymiarowości danych). Istnieje również funkcja *TREE* generująca dendrogram na podstawie wyników metod *CLUSTER* lub *VARCLUS*. Do grupowania dużych zbiorów danych producent zaleca wykorzystanie metody *FASTCLUS* prawdopodobnie ze względu na niższą niż w pozostałych przypadkach złożoność obliczeniową zastosowanego algorytmu k-średnich.

IBM SPSS Modeler

Kolejne omawiane narzędzie komercyjne rozwijane aktualnie przez firmę IBM nosi nazwę SPSS Modeler i w kwestii interfejsu użytkownika jest bardzo podobne do poprzedniego rozwiązania (SAS Enterprise Miner). Tutaj również wykorzystywana jest technika przeciągnij i upuść podczas projektowania kolejnych etapów procesu analizy danych. Jednakże produkt firmy IBM jest w pełni autonomiczny i nie wymaga żadnej dodatkowej infrastruktury do swojego działania (co nie jest prawdą w przypadku konkurencji z SAS Institute, ponieważ wymagane jest zainstalowanie kilku dodatkowych komponentów z którymi komunikuje się Enterprise Miner). Istnieje oczywiście możliwość korzystania z niego w środowisku rozproszonym, wykorzystując komplementarny SPSS Modeler Server, dzięki czemu obliczenia wykonywane są na dedykowanym serwerze (lub grupie serwerów), a użytkownikowi prezentowane są wyniki przetwarzania, którymi może dowolnie manipulować na komputerze lokalnym. Dodatkowo SPSS Modeler wykorzystuje menu typu wstążka (ang. *ribbon*) znane chociażby z nowszych wersji pakietu biurowego Microsoft Office, dzięki czemu poszczególne elementy, z których składa się proces analizy (jak wczytywanie danych z określonych źródeł, ich transformacja, prezentacja graficzna na wykresach, modelowanie, eksport wyników) są intuicyjnie posegregowane w odpowiednich zakładkach. Ułatwia to znacząco korzystanie z tego oprogramowania, nawet bez konieczności zaglądania do dokumentacji. Przykładowy zrzut ekranu prezentujący wygląd i działanie opisywanego programu został zilustrowany na rysunku 3.4. Najnowsza wersja jest oznaczona numerem 15 (dostępna od czerwca 2012).

[6] Proces graficznej eksploracji danych został omówiony w rozdziale 6.



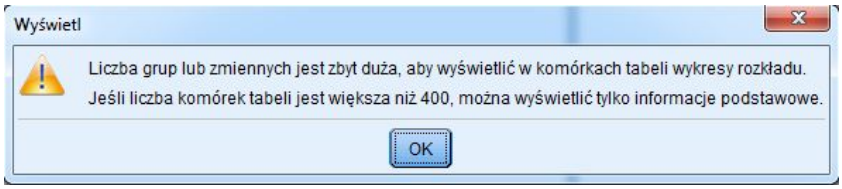
Rysunek 3.4: Interfejs programu IBM SPSS Modeler.
Źródło: Opracowanie własne

W oficjalnej dokumentacji [81] (sekcja Algorithms Guide) wymienionych jest 35 algorytmów[7] modelowania i analizy danych wśród których znajdują się metody wykrywania odchyleń, selekcji cech, generowania drzew decyzyjnych, grupowania czy odkrywania sekwencji. Cechą wyróżniającą IBM SPSS Modeler jest możliwość integracji z kilkoma rozwiązaniami eksploracji danych, implementowanej bezpośrednio na serwerze bazodanowym (łącząc się przykładowo z Microsoft Analysis Services omawianym w dalszej sekcji). Pozwala to na budowę, ocenę i przechowywanie utworzonych modeli bezpośrednio na serwerze bazy danych. Dzięki temu można połączyć analityczne możliwości oprogramowania SPSS Modeler (jak na przykład generowanie wykresów rozrzutu czy statystyk opisowych) z wydajnością i wyspecjalizowanymi algorytmami zarządzania danymi poszczególnych producentów relacyjnych systemów bazodanowych. Szczegółowy opis wspomnianego rozwiązania znajduje się w [81] (sekcja In-Database Mining Guide).

Możliwości przeprowadzenia procesu analizy skupień ograniczają się w omawianym programie do czterech algorytmów grupowania: k-średnich, mapy samoorganizującej się Kohonena, dwupoziomowego grupowania oraz wykrywania odchyleń. W przypadku niehierarchicznego al-

[7] Niektóre z wymienionych pozycji stanowią grupę algorytmów realizujących to samo zadanie np. wykrywanie odchyleń.

gorytmu k-średnich oraz mapy samoorganizującej, mimo zastosowania miary euklidesowej jako podobieństwa dwóch obiektów, możliwe jest przetwarzanie zarówno danych jakościowych oraz ilościowych. Dane jakościowe kodowane są jako wektory numeryczne o długości odpowiadającej liczbie wartości przyjmowanych przez określony atrybut. Algorytm dwupoziomowego grupowania składa się z dwóch faz: wstępnej polegającej na analizie rekordów zbioru jeden po drugim i ich łączeniu w jedno skupienie (jeżeli odpowiedni próg podobieństwa między nimi jest spełniony) oraz faktycznego grupowania techniką (hierarchiczną) aglomeracyjną dopóki nie zostanie utworzona ustalona przez użytkownika liczba skupień. Metoda wykrywania odchyleń jest tak naprawdę zastosowaniem opisanej idei dwupoziomowego grupowania wraz z generowaniem statystyk opisu grup (m.in. bazując na notacji odchylenia standardowego) i ich klasyfikacji jako grupy obiektów odstających.



Rysunek 3.5: Problem z wyświetlaniem wyników grupowania w IBM SPSS Modeler.
Źródło: Opracowanie własne

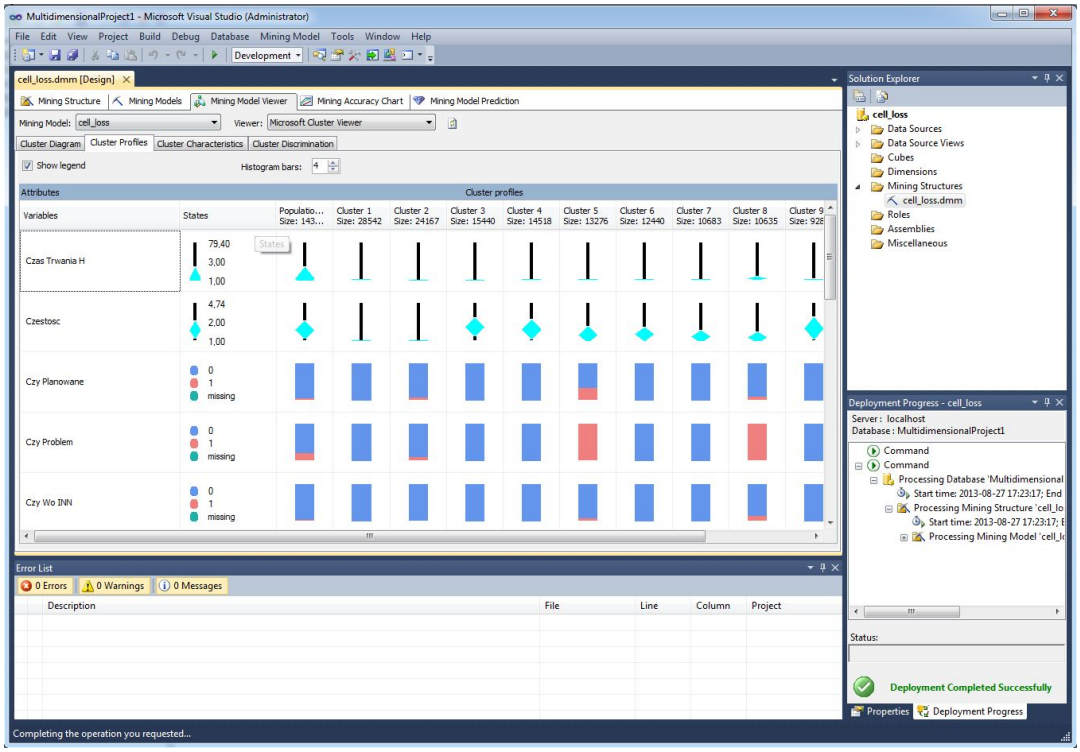
Niestety wspomniane oprogramowanie nie jest najlepiej dostosowane do przetwarzania dużych zbiorów danych złożonych. Przy próbie zastosowania algorytmu k-średnich do omawianego w rozdziale 2 zbioru *cell_loss* w celu wygenerowania 500 skupień[8] program wyświetla komunikat, odnośnie zbyt dużej liczby grup lub zmiennych by przedstawić wszystkie wyniki, zaprezentowany na rysunku 3.5. Dzieje się tak ponieważ oprogramowanie wyświetla domyślnie rezultaty grupowania korzystając z macierzy wykresów rozrzutu dla wszystkich możliwych kombinacji cech. W przypadku, gdy jest ich zbyt dużo (cech lub grup), wyświetlane są jedynie podstawowe informacje o utworzonym podziale na skupienia, takie jak liczba utworzonych grup, ich rozmiary czy jakość uzyskanej struktury wykorzystując przykładowo współczynnik sylwetki (ang. *silhouette coefficient*)[9]. Brak jest również bardziej wyspecjalizowanych algorytmów analizy skupień (jak podejścia gęstościowe omawiane w rozdziale 5).

Microsoft Analysis Services

Microsoft Analysis Services jest komponentem wchodzącym w skład instalacji serwera bazy danych SQL Server, służącym do bezpośredniego przetwarzania analitycznego (ang. ***On**Line **A**nalytical **P**rocessing*), eksploracji danych i generowania raportów. Nie jest to zatem narzędzie dedykowane przede wszystkim wykrywaniu zależności i wzorców jak poprzednie. Jest to związane z historią powstania tego oprogramowania. W 1996 roku Microsoft przejął technologię przetwarzania OLAP z izraelskiej firmy Panama Software i dwa lata później stała się ona

[8] Liczba skupień została dobrana arbitralnie, wyłącznie w celu przetestowania możliwości oprogramowania.
[9] Problematyka oceny jakości grupowania została szczegółowo opisana m.in. w [21, 19], jak również został poświęcony jej odrębny rozdział pracy magisterskiej autora [71].

integralną częścią ich bazodanowego serwera [82]. Następnie w 2000 roku zostały do tego produktu dodane metody eksploracji danych, zachowując jednocześnie wsteczną kompatybilność z poprzednią wersją. Należy jednak wyraźnie rozróżnić ideę OLAP od analizy eksploracyjnej. Eksploracja danych ma na celu odkrycie nowych trendów, korelacji czy zależności w dużych zbiorach danych, natomiast bezpośrednie przetwarzanie analityczne najczęściej służy jako narzędzie do wspomagania podejmowania strategicznych decyzji biznesowych. Przykładowo za pomocą OLAP użytkownik może w szybkim tempie przeanalizować wyniki sprzedaży w danym przedsiębiorstwie z podziałem na zyski, obroty, liczbę transakcji dla określonych lokalizacji, uwzględniając segregację na poszczególnych pracowników w odniesieniu do konkretnych lat. Wyniki tego typu badań mogą służyć podjęciu decyzji o zwiększeniu kampanii marketingowej bądź zmianie polityki personalnej danej firmy celem zwiększenia zysków. Generalizując można stwierdzić, że eksploracja danych odkrywa zależności i hipotezy, które dzięki mechanizmowi OLAP można potwierdzić. Bardziej szczegółowe informacje na temat bezpośredniego przetwarzania analitycznego i eksploracji danych można znaleźć w książce [23].



Rysunek 3.6: Interfejs programu do wydobywania wiedzy za pomocą Microsoft Analysis Services.

Źródło: Opracowanie własne

Interfejs programu w najnowszej wersji 2012 został przedstawiony na rysunku 3.6. Jest on oparty o silnik środowiska programistycznego Visual Studio, co ułatwia poruszanie się i korzystanie z dostępnych funkcjonalności. W prawej stronie okna znajduje się drzewo katalogów, w którym można tworzyć m.in. nowe źródła danych (zarówno pochodzące z bazy jak i plików tekstowych różnego formatu), widoki danych (pozwalające wybrać wyłącznie interesujące ta-

bele lub dokonać ich złączenia), kostki (pozwalające przeprowadzić wielowymiarową analizę OLAP), modele eksploracji danych. Główną część obszaru roboczego zajmują wyniki danego procesu analizy danych. W zaprezentowanym na rysunku 3.6 przypadku, dokonano grupowania zbioru *cell_loss* na 10 skupień[10]. Cechą wyróżniającą rozwiązanie firmy Microsoft od pozostałych jest ciekawa koncepcja wizualizacji grup w formie tabeli, której kolumny symbolizują wygenerowane skupienia, a wiersze to poszczególne cechy brane pod uwagę w tym procesie. Na przecięciu wiersza i kolumny generowane są wykresy słupkowe bądź pudełkowe (w zależności od typu danych), które obrazują rozkład wartości danej cechy w określonym skupieniu. Możliwe jest również wygenerowanie grafu powiązań między skupieniami (na podstawie ich wzajemnego podobieństwa). Jest to jedno z nielicznych rozwiązań programowych, które umożliwiło bezproblemową wizualizację 500 skupień[11]. Nie bez znaczenia jest również ścisła integracja z serwerem bazy danych, dzięki czemu operacje tworzenia modelu wykorzystują struktury indeksowane (jak np. B-drzewa) i wyspecjalizowane algorytmy dostępowe do dysku twardego, przyspieszając znacząco generowanie wyników[12].

Wadą omawianego oprogramowania jest mała dostępność algorytmów analizy skupień. Zgodnie z oficjalną dokumentacją techniczną [83] zostały zaimplementowane wyłącznie dwa algorytmy grupowania: k-średnich oraz EM (ang. ***E**xpectation–**M**aximization algorithm*). Jednakże oba algorytmy zostały tak dostosowane, by móc operować zarówno na danych ilościowych jak i jakościowych. Sama implementacja nie narzuca żadnych ograniczeń co do oczekiwanej liczby wygenerowanych grup, a maksymalna liczba poszczególnych cech (i ich unikalnych wartości) została ustawiona na ponad 65 tysięcy. Głównym czynnikiem limitującym możliwości oprogramowania wydaje się dopuszczalny czas na dokonanie analiz (grupowania). Szczegóły dotyczące parametrów jak również ogólnej zasady działania oraz konkretnej implementacji obu algorytmów analizy skupień są udostępniane w dokumentacji opisywanego narzędzia [83].

Kolejna wada (przy korzystaniu ze środowiska opartego o Visual Studio) to brak graficznych metod opisu danych w postaci wykresów rozrzutu czy dowolnie tworzonych histogramów. Nie ma możliwości dokonania pełnej, eksploracyjnej analizy danych zgodnej z definicją J. Tukey'a[13]. Sytuację nieco poprawia możliwość wykorzystania środowiska MS Excel do analizy danych. Wówczas Microsoft Analysis Services jest tylko usługą udostępniającą różne techniki analizy skupień, natomiast dodatkowo można skorzystać ze wszystkich funkcjonalności tworzenia wykresów i wizualizacji danych oferowanych przez Excel'a. Mimo wszystko, w opinii autora, rozwiązania konkurencyjne jak (STATISTICA czy IBM SPSS Modeler) oferują pod tym względem więcej możliwości.

[10] Wybór małej liczby skupień podyktowany był wyłącznie przejrzystością zrzutu ekranu działającego oprogramowania.

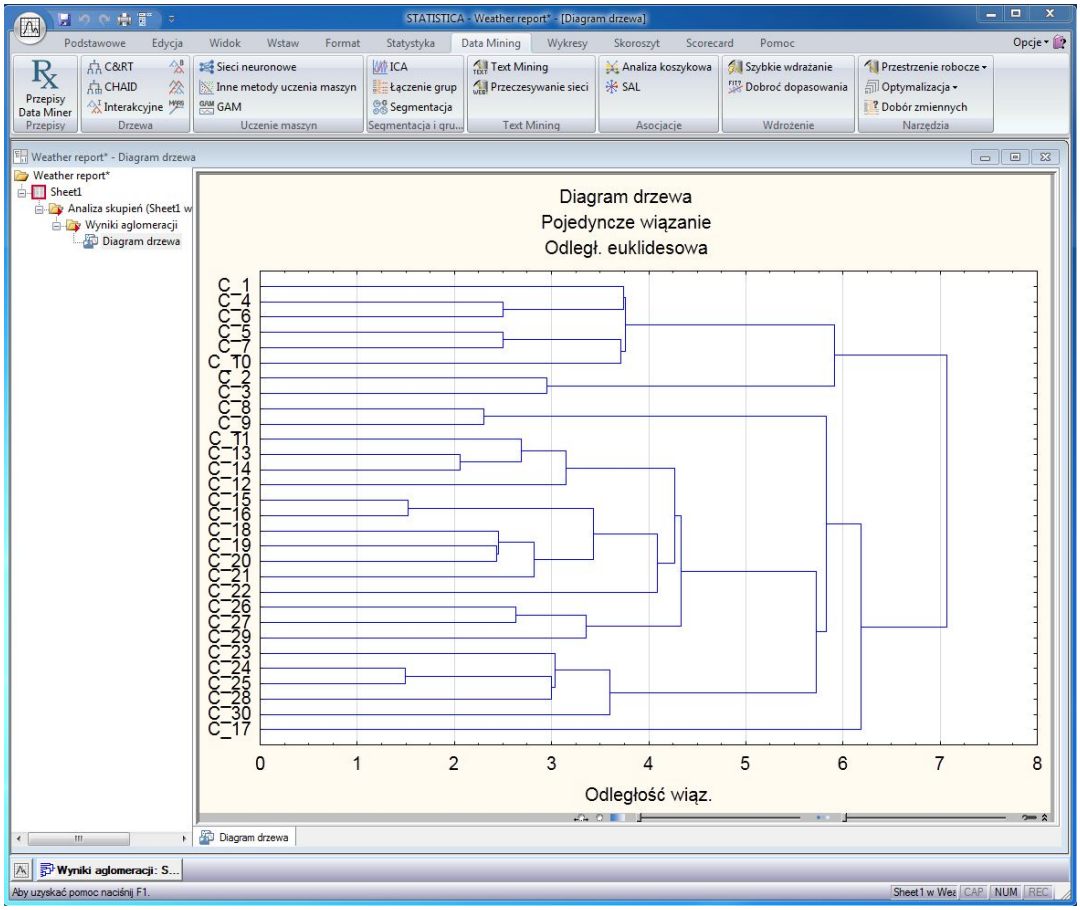
[11] Liczba 500 została dobrana arbitralnie, jednakże stanowi dobry przypadek testowy dla oprogramowania eksploracyjnej analizy skupień.

[12] Bardziej szczegółowe informacje odnośnie organizacji bazy danych, struktur indeksowanych oraz algorytmów dostępu do dysku znajdują się w [23].

[13] Pojęcie eksploracyjnej analizy danych zostało szczegółowo omówione w rozdziale 4.

STATISTICA Data Miner

Ostatnie komercyjne narzędzie do wydobywania wiedzy, analizowane w niniejszej pracy, to STATISTICA Data Miner firmy StatSoft. Interakcja z programem odbywa się przez wykorzystanie stosownych kreatorów (prowadzących użytkownika przez wszystkie etapy analizy, od ustalenia źródła i przygotowania danych, przez modyfikację parametrów do wygenerowania wybranego modelu, na raportowaniu wyników skończywszy) lub bezpośrednio wybierając odpowiedni przycisk i zakładkę na interfejsie wstążki. Zatem w przeciwieństwie do produktów SAS Institute czy IBM, proces wydobywania wiedzy nie przypomina tworzenia grafu (którego węzły reprezentują poszczególne jego etapy), a bardziej skupia się na pojedynczych czynnościach, których wyniki wyświetlają się w osobnych oknach. Wydaje się to rozsądnym podejściem, ponieważ nie zawsze możliwe jest zaprojektowanie od razu wszystkich kroków procesu odkrywania wiedzy – decyzje odnośnie dalszego postępowania mogą być uzależnione od wyników poprzednich kroków. Interfejs programu w wersji 10 został zaprezentowany na rysunku 3.7.

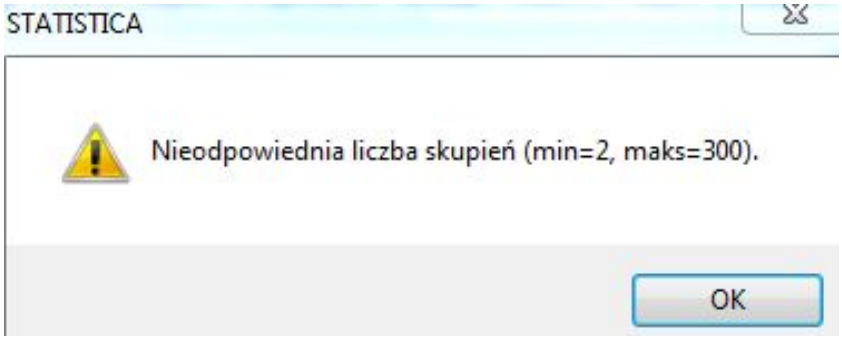


Rysunek 3.7: Interfejs programu STATISTICA Data Miner.
Źródło: Opracowanie własne

Producent opisując możliwości produktu chwali się najszerszym wyborem metod eksploracji danych dostępnych na rynku "np. bardzo rozbudowany zestaw technik analizy skupień

(segmentacji), architektur sieci neuronowych, drzew klasyfikacyjnych i regresyjnych, modelowania wielowymiarowego (w tym MARSplines) i wielu innych technik predykcyjnych, szeroka gama narzędzi graficznych" [78]. Ponadto bardzo silnie akcentowane są czynniki dostosowania programu do przetwarzania olbrzymich zbiorów danych, zawierających miliony obserwacji i cech, zoptymalizowany dostęp do baz danych oraz możliwość skorzystania z wielu narzędzi graficznych (jako nawiązanie do wizualnych metod opisu danych). Jednakże możliwość przetwarzania olbrzymich zbiorów danych, została uzyskana przez dobór i eliminację cech (wybór jedynie pewnego podzbioru atrybutów) w przypadku baz o dużej liczbie kolumn, natomiast do przetwarzania "danych o praktycznie dowolnej liczbie przypadków" [78] wykorzystywane jest próbkowanie losowe, wybierające reprezentatywny podzbiór obiektów. Poza tym możliwe jest zastosowanie przetwarzania rozproszonego w architekturze klient-serwer, co przenosi całe przetwarzanie na szereg wyznaczonych do tego maszyn serwerowych, a komputer klienta pełni rolę terminala wyświetlającego wyniki przetwarzania i interfejsu programu w przeglądarce internetowej. Dzięki temu osiągnięto również pewną niezależność od architektury sprzętowej i używanego systemu operacyjnego.

STATISTICA Data Miner udostępnia trzy algorytmy analizy skupień: k-średnich, oczekiwania-maksymalizacji (ang. ***E**xpectation-**M**aximization*) oraz aglomeracyjny. Zdolność metody k-średnich do grupowania danych ilościowych i jakościowych warunkowana jest wyborem miary odległości. Dostępne są odległości: euklidesowa, Czebyszewa, miejska, potęgowa, kwadrat odległości euklidesowej oraz niezgodność procentowa [86]. W przypadku algorytmu hierarchicznego (aglomeracyjnego) zaimplementowane zostały następujące metody łączenia skupień: pojedynczego i pełnego wiązania, metoda średnich połączeń (również ważonych), środków ciężkości, mediany oraz Warda [19]. Nie bez znaczenia jest również możliwość grupowania cech i obiektów jednocześnie (lub autonomicznie). Mimo wszystko widoczny jest brak bardziej zaawansowanych metod analizy skupień (wykorzystujących pojęcie gęstości i innych). Ponadto algorytm k-średnich posiada ograniczenie maksymalnej liczby generowanych grup do 300. Próba ustawienia większej wartości (co może być uzasadnione w przypadku analizy rzeczywistych zbiorów złożonych) skutkuje wyświetleniem komunikatu o błędzie zaprezentowanym na rysunku 3.8.



Rysunek 3.8: Komunikat informujący o ograniczeniu maksymalnej liczby generowanych grup.
Źródło: Opracowanie własne

Wspomniane wady w zakresie analizy skupień, oprogramowanie STATISTICA Data Mi-

ner nadrabia oferując szerokie spektrum statystyk opisowych, wśród których można wyróżnić korelacje liniowe i nieliniowe, test t-studenta, tabele liczności (w tym wielorozdzielcze). Nie bez znaczenia jest również szeroki wybór technik graficznych analizy danych jak histogramy, wykresy pudełkowe, rozrzutu, kwartyl-kwartyl, przestrzenne, obrazkowe czy macierzowe [86]. Jest to bez wątpienia największy zestaw technik graficznych wśród omawianych narzędzi komercyjnych, przez co to właśnie STATISTICA Data Miner jest w opinii autora najlepszym obecnie dostępnym komercyjnym narzędziem do wydobywania wiedzy z danych. Opinię tę zdaje się potwierdzać rosnąca popularność tego oprogramowania w ankiecie zilustrowanej na rysunku 3.1.

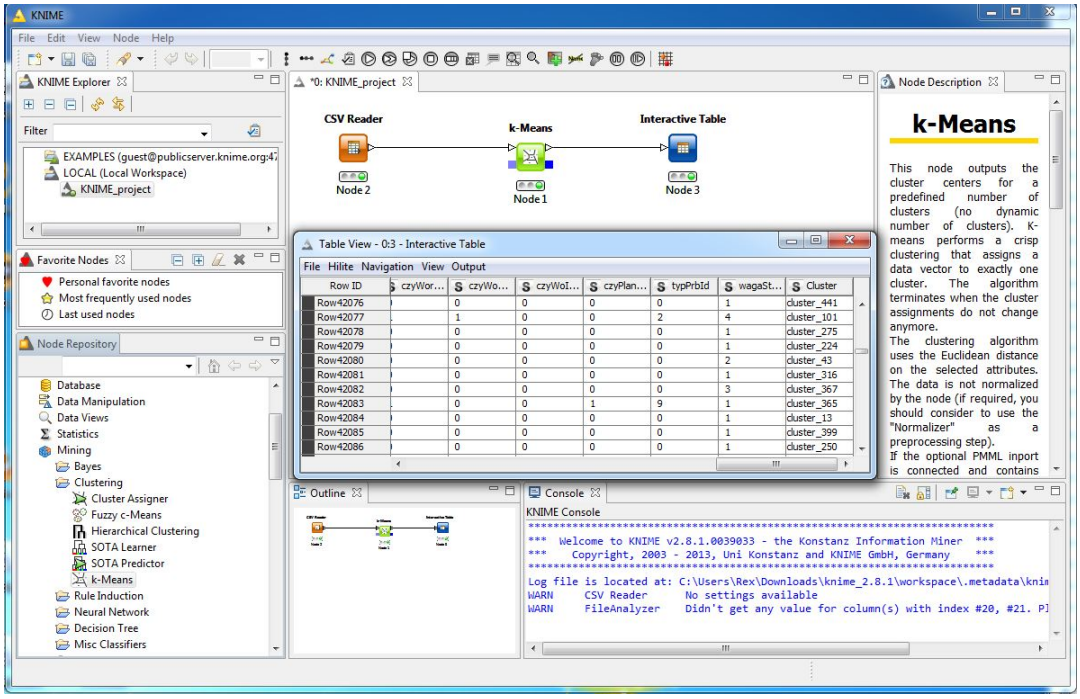
3.2 Oprogramowanie niekomercyjne

Wszystkie omawiane poniżej przykłady programów niekomercyjnych są zgodne z ideą otwartego oprogramowania co oznacza, że dostępny jest ich kod źródłowy oraz oparte są o mało restrykcyjne licencje, które umożliwiają dokonywanie zmian w sposobie ich działania jak również ewentualną integrację z własnymi rozwiązaniami. Przeprowadzony w niniejszej pracy przegląd możliwości takich programów pozwoli odpowiedzieć na pytanie, czy mogą one rywalizować z płatnymi rozwiązaniami komercyjnymi.

KNIME

KNIME (ang. ***Ko****N**stanz* ***I****nformation* ***M****in****E****r*) jest modularnym oprogramowaniem do wydobywania wiedzy z danych, opartym o interfejs środowiska programistycznego Eclipse. Sposób posługiwania się programem jest bardzo podobny do rozwiązania zastosowanego w SAS Enterprise Miner czy IBM SPSS Modeler i polega na łączeniu w graf elementów (zwanych węzłami) symbolizujących ładowanie danych, ich transformację, wizualizację, generowanie statystyk, modelowanie, zapisywanie wyników jak zaprezentowano na rysunku 3.9. Architektura całego systemu została zrealizowana przy uwzględnieniu trzech głównych zasad: interaktywnego środowiska graficznego, modularności i rozszerzalności [9]. Cele te osiągnięto implementując mechanizm przeciągnij i upuść, dzięki czemu zarządzanie czy zmiana poszczególnych etapów wydobywania wiedzy jest bardzo intuicyjna. Ponadto struktury danych wykorzystywane w zaimplementowanych algorytmach są od siebie niezależne (dzięki zasadzie hermetyzacji znanej z programowania obiektowego), przez co można tworzyć niemal dowolne kombinacje połączeń wśród dostępnych węzłów. Możliwość rozszerzenia istniejącej funkcjonalności (bez konieczności manualnego kopiowania bibliotek bądź korzystania z instalatorów) uzyskano integrując zarządcę pakietów (jak w środowisku Eclipse), który pobiera ze zdalnego repozytorium nowe dodatki, a zadaniem użytkownika jest jedynie dokonanie wyboru jaki element chce dołączyć i zaakceptowanie decyzji. Zarządca pakietów autonomicznie pobierze i wprowadzi niezbędne zmiany w konfiguracji programu.

Możliwości omawianego oprogramowania w kategorii analizy skupień domyślnie ograniczają się do algorytmów: k-średnich, fuzzy c-means, hierarchicznego (zarówno aglomeracyjnego jak i deglomeracyjnego) oraz SOTA Lerner. W przypadku klasycznego algorytmu k-średnich wykorzystywana jest wyłącznie odległość euklidesowa (w odniesieniu do danych ilościowych).



Rysunek 3.9: Interfejs programu KNIME.
Źródło: Opracowanie własne

Należy jednak zaznaczyć, że dostępny jest osobny węzeł (komponent) o nazwie Distance Matrix Calculate, którego zadaniem jest wyznaczenie macierzy niepodobieństwa, z użyciem miar takich jak: odległość euklidesowa, miejska, Tanimoto, korelacji kosinusowej, czy współczynnika Dice'a. Dodatkowo bez większych problemów (z ograniczeniem jednak do cech ilościowych) generuje on grupowanie dla zbioru danych rzeczywistych *cell_loss* przy ustalonej jako 500 liczbie skupień (co nie było możliwe we wszystkich produktach komercyjnych). Dla algorytmu hierarchicznego zaimplementowano trzy standardowe miary łączenia skupień: pojedynczego, średniego oraz całkowitego wiązania, a dostępne miary niepodobieństwa (odległości) to odległość euklidesowa i miejska. Metoda fuzzy c-means działa analogicznie do algorytmu k-średnich, z tą różnicą, że dany obiekt nie musi należeć wyłącznie do jednego skupienia – jest to tzw. grupowanie rozmyte. Węzeł SOTA Lerner to algorytm generujący hierarchię skupień na podstawie budowy tzn. drzewa samoorganizującego się (ang. ***S**elf-**O**rganising **T**ree **A**lgorithm*). Jest to sieć neuronowa, zorganizowana zgodnie z topologią drzewa binarnego, często wykorzystywana przy analizie danych mikromacierzowych[14].

Jak zaznaczono wcześniej, jedną z istotnych cech opisywanego oprogramowania jest jego rozszerzalność. Dzięki temu do ogólnie dostępnej palety technik analizy skupień można dołączyć algorytm k-medoidów oraz wszystkie dostępne w systemie Weka (opisywanym w dalszej części tego rozdziału). Po zainstalowaniu dodatkowych pakietów, użytkownik uzyskuje zatem dostęp m. in. do algorytmów gęstościowych DBSCAN i OPTICS[15]. Niestety sposób integracji

[14] Szczegóły na temat budowy i zasady działania wspomnianego algorytmu można znaleźć w [24].
[15] Analizie algorytmów gęstościowych został poświęcony rozdział 5.

z innymi rozwiązaniami do eksploracji danych to również największa wada opisywanego programu. Wyniki działania dodatkowych algorytmów analizy skupień wyświetlane są w innym oknie i interfejsie zgodnym z oprogramowaniem, z którego zostały zapożyczone. Implikuje to również niemożność uwzględnienia tych rezultatów, w dalszych etapach procesu wydobywania wiedzy projektowanego w KNIME – wyjścia algorytmów zapożyczonych z pakietu Weka nie da się połączyć z innymi elementami (węzłami) dostępnymi w KNIME. Nie jest możliwe zatem np. pobranie przydziału do grup oraz wizualizacja ich rozmiarów na wykresie kołowym. KNIME pełni zatem pewnego rodzaju warstwę pośredniczącą, która dokonuje załadowania i transformacji danych, a następnie uruchamia procedury innego oprogramowania generując w nim wyniki (podobnie jak zrobiłby to użytkownik korzystając bezpośrednio z Weki).

Metody opisu i wizualizacji danych wśród których można wyróżnić wykresy pudełkowe, rozrzutu, radarowe, histogramy, technikę współrzędnych równoległych można dodatkowo rozszerzyć o możliwości reprezentacji graficznej biblioteki JFreeChart [80] i wszystkich komend dostępnych w pakiecie R. Jest to zatem bardzo rozbudowane i modyfikowalne oprogramowanie, które z powodzeniem może konkurować z rozwiązaniami komercyjnymi. Szczegóły na temat możliwości KNIME (w szczególności wizualizacyjnych i statystycznych) znajdują się w książce jego pomysłodawców poświęconej eksploracji danych [8].

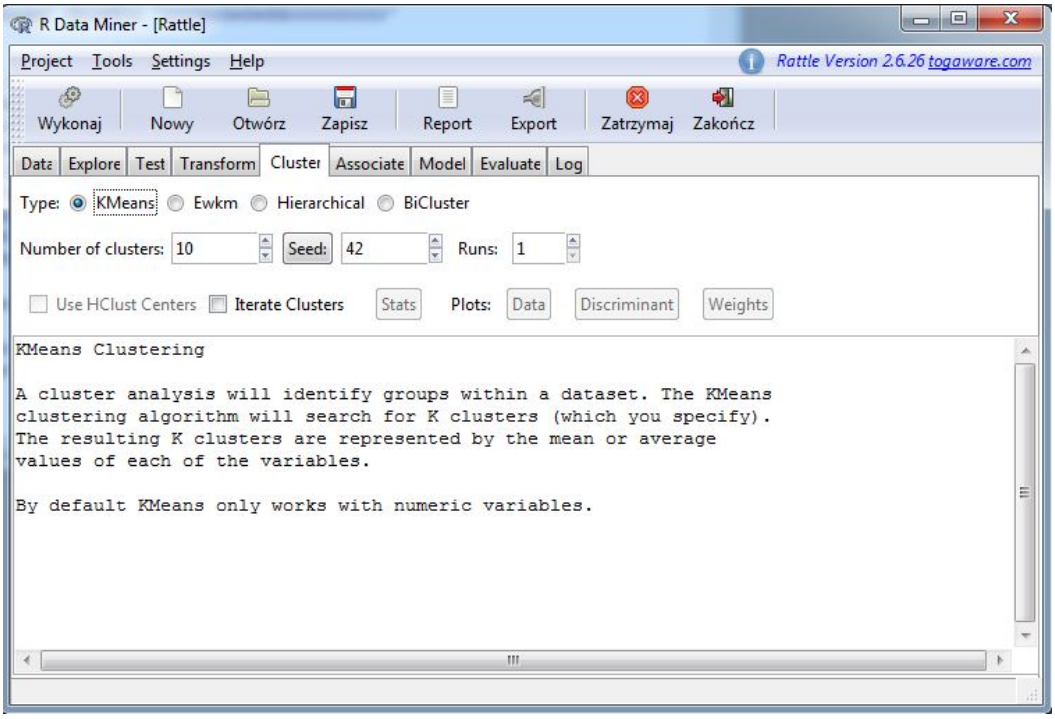
R z nakładką Rattle

Rattle (ang. ***R** **A**nalytical **T**ool **T**o **L**earn **E**asily*) jest graficzną nakładką (z dodatkowymi funkcjami) na znany w środowisku analityków pakiet statystyczny R. Pozwala na ładowanie danych z wielu źródeł (w tym bazy danych, plików tekstowych, arkuszy kalkulacyjnych, natywnych formatów wykorzystywanych przez R czy Weka), ich wstępną eksplorację (na podstawie statystyk opisowych jak średnia arytmetyczna, odchylenie standardowe oraz technik graficznych typu histogramy czy wykresy pudełkowe), transformację (poprzez normalizację czy usuwanie wartości pustych), tworzenie modeli (jak skupienia czy reguły asocjacyjne) oraz ewaluację (wykorzystując przykładowo krzywe ROC) [89]. Interfejs programu w wersji 2.6.26 został przedstawiony na rysunku 3.10 i składa się z jednego okna podzielonego na szereg zakładek. Zakładki powinny być przetwarzane kolejno (wykonując np. transformację i kodowanie danych przed zastosowaniem metody generowania reguł asocjacyjnych) lub autonomicznie, jeżeli użytkownika interesuje wyłącznie zastosowanie konkretnej techniki eksploracji.

Z poziomu zakładki Cluster dostępne są cztery algorytmy analizy skupień: k-średnich, Ewkm, hierarchiczny oraz BiCluster[16] (służący do grupowania zarówno obiektów jak i cech jednocześnie). Technika k-średnich ograniczona jest do przetwarzania danych ilościowych, jednakże potrafi wygenerować dla zbioru *cell_loss*, uznaną jako testową, liczbę 500 skupień. Ewkm jest wariantem metody k-średnich, przyporządkowującym wagi cechom uznanym za istotne, podczas wyznaczania podobieństwa dwóch obiektów[17]. Niestety algorytm generujący hierarchię nie może zostać zastosowany do zbioru danych rzeczywistych *cell_loss* i większych, ponieważ

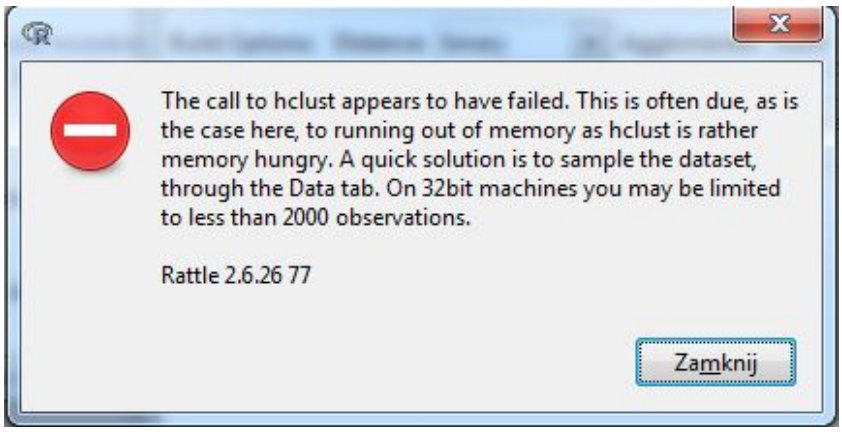
[16] Szczegóły na temat działania i zastosowania algorytmu BiCluster dostępne są pod adresem http://cran.r-project.org/web/packages/biclust/biclust.pdf.

[17] Szczegóły na temat działania i zastosowania algorytmu Ewkm dostępne są pod adresem http://cran.r-project.org/web/packages/weightedKmeans/weightedKmeans.pdf.



Rysunek 3.10: Graficzny interfejs Rattle.
Źródło: Opracowanie własne

oprogramowanie wyświetla komunikat, o niewystarczającej ilości dostępnej pamięci operacyjnej i zbyt dużej liczbie obiektów w zbiorze[18], zaprezentowany na rysunku 3.11. Proponowane rozwiązanie tego problemu to wykorzystanie metody próbkowania (i operowanie wyłącznie na dużo mniejszym podzbiorze obiektów).



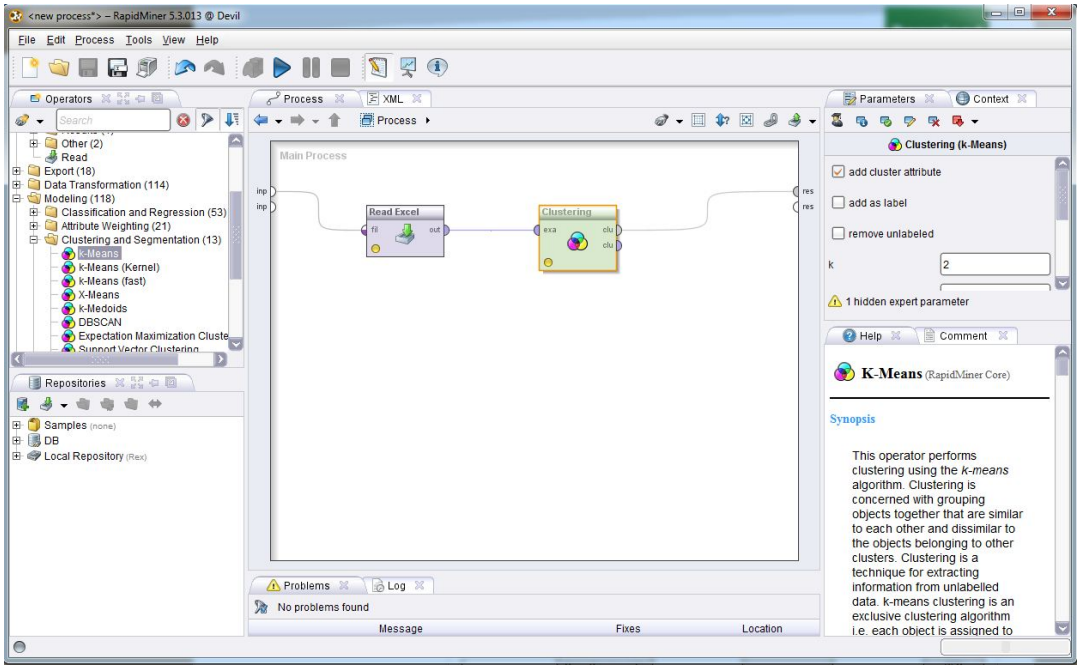
Rysunek 3.11: Komunikat informujący o ograniczeniu algorytmu hierarchicznego.
Źródło: Opracowanie własne

[18] Analiza omówionych w tym rozdziale narzędzi wydobywania wiedzy odbywała się przy wykorzystaniu systemu Windows 7 w wersji 64 bitowej oraz komputera wyposażonego w procesor core i5 661 3,33 GHz i 8 GB pamięci RAM.

Dostępne w Rattle algorytmy nie powinny być zatem bezpośrednio zastosowane przy grupowaniu rzeczywistych zbiorów danych złożonych. Małą liczbę algorytmów można rozszerzyć instalując, poprzez konsolę oprogramowania R, pakiet *fpc*[19], który dostarcza sześć dodatkowych metod analizy skupień, wśród których znajduje się również technika gęstościowa DBSCAN. Brak jest jednak (proponowanej w dalszych rozdziałach, do zastosowania przy zadaniu wydobywania wiedzy) metody OPTICS (której implementacja w R jest poszukiwana na wielu grupach dyskusyjnych). Niestety posługiwanie się algorytmem DBSCAN możliwe jest wyłącznie przez linię komend R, co wymaga stosownej wiedzy na temat syntaktyki poleceń i poruszania się w środowisku konsolowym. Podobnie jest w przypadku generowania niedostępnych przez Rattle wykresów czy technik wizualizacyjnych. Szczegóły odnośnie pełnych możliwości współpracy nakładki Rattle oraz pakietu R prezentuje książka [70].

RapidMiner

RapidMiner jest środowiskiem do wydobywania wiedzy z danych, które posiada zarówno wersję darmową (na licencji AGPL udostępniającą kod źródłowy) oraz szereg wariantów komercyjnych. Poszczególne wersje różnią się m.in. możliwościami ładowania danych (przykładowo w wersjach komercyjnych jest możliwość podłączenia się do narzędzia SAP), obsługą wielu rdzeni procesora, dostępnością edytora danych oraz prowadzeniem procesu eksploracji bezpośrednio na silniki bazy danych (podobnie jak IBM SPSS Modeler). W tej sekcji zostanie przeanalizowana wersja niekomercyjna o numerze 5.3.0.13.

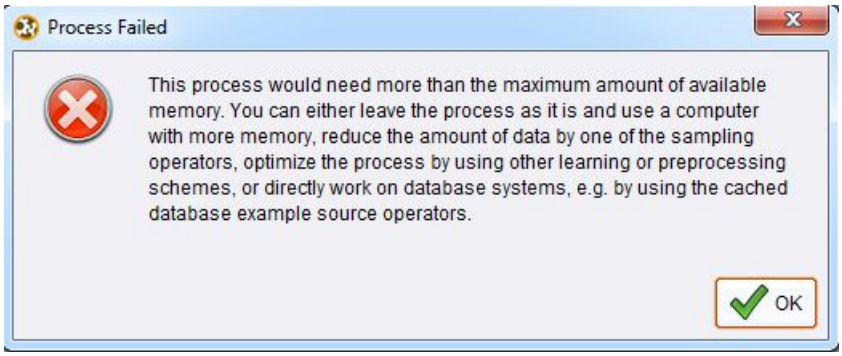


Rysunek 3.12: Interfejs programu RapidMiner.
Źródło: Opracowanie własne

[19]Szczegółowy opis możliwości pakietu *fpc* znajduje się pod adresem http://cran.r-project.org/web/packages/fpc/fpc.pdf.

Interfejs omawianego programu (zaprezentowany na rysunku 3.12) jest bardzo podobny do rozwiązania stosowanego w KNIME, gdzie dostępnych jest szereg mniejszych okien (zintegrowanych z głównym) podzielonych na zakładki, które można dowolnie przemieszczać. Również projektowanie przebiegu procesu odkrywania wiedzy wygląda podobnie – z okna w lewym górnym rogu ekranu wybiera się tzw. operatory (będące analogią do węzłów KNIME) czyli elementy realizujące ładowanie danych, transformacje, modelowanie, wizualizację oraz inne, a następnie łączy się operatory w graf skierowany, korzystając z centralnie ustawionego okna obszaru roboczego. Każdy operator ma oczywiście unikalne właściwości sterujące jego pracą (w zależności od pełnionej funkcji może to być ścieżka i format pliku do odczytu bądź wymagana liczba skupień w przypadku algorytmu grupowania).

RapidMiner posiada 11 zaimplementowanych algorytmów analizy skupień, wśród których można wyróżnić: cztery warianty metody k-średnich, technikę k-medoids, DBSCAN, EM, metodę wektorów nośnych SVC (ang. ***S**upport **V**ector **C**lustering*), grupowanie hierarchiczne (aglomeracyjne i deglomeracyjne) oraz przyporządkowanie losowe[20]. Należy nadmienić, że dla algorytmu DBSCAN dostępna jest miara nazywana *Mixed Euclidean*, która jest modyfikacją miary euklidesowej umożliwiającą porównywanie danych ilościowych i jakościowych – dla cech nominalnych miara przyporządkowuje wartość zero lub jeden w zależności od tego czy wartości (dla dwóch porównywanych obiektów) są sobie równe. Algorytm jednakże zgłasza błąd, gdy w załadowanym zbiorze występują wartości puste. Przy przetwarzaniu dużych zbiorów danych może również dojść do problemów pamięciowych – program przechowuje wszystkie dane w pamięci operacyjnej, dlatego w zależności od złożoności procesu odkrywania wiedzy może pojawić się komunikat (jak zaprezentowano na rysunku 3.13) informujący o konieczności zmniejszenia objętości analizowanego zbioru lub powiększenia zasobów wolnej pamięci RAM. Taka sytuacja miała miejsce podczas próby zastosowania algorytmu k-średnich do rzeczywistego zbioru danych *cell_loss*. Wśród dostępnych algorytmów brak jest jednakże techniki OPTICS.



Rysunek 3.13: Komunikat o zbyt małej ilości dostępnej pamięci.
Źródło: Opracowanie własne

Możliwości wizualizacyjne[21] oprogramowania obejmują m.in. generowanie histogramów,

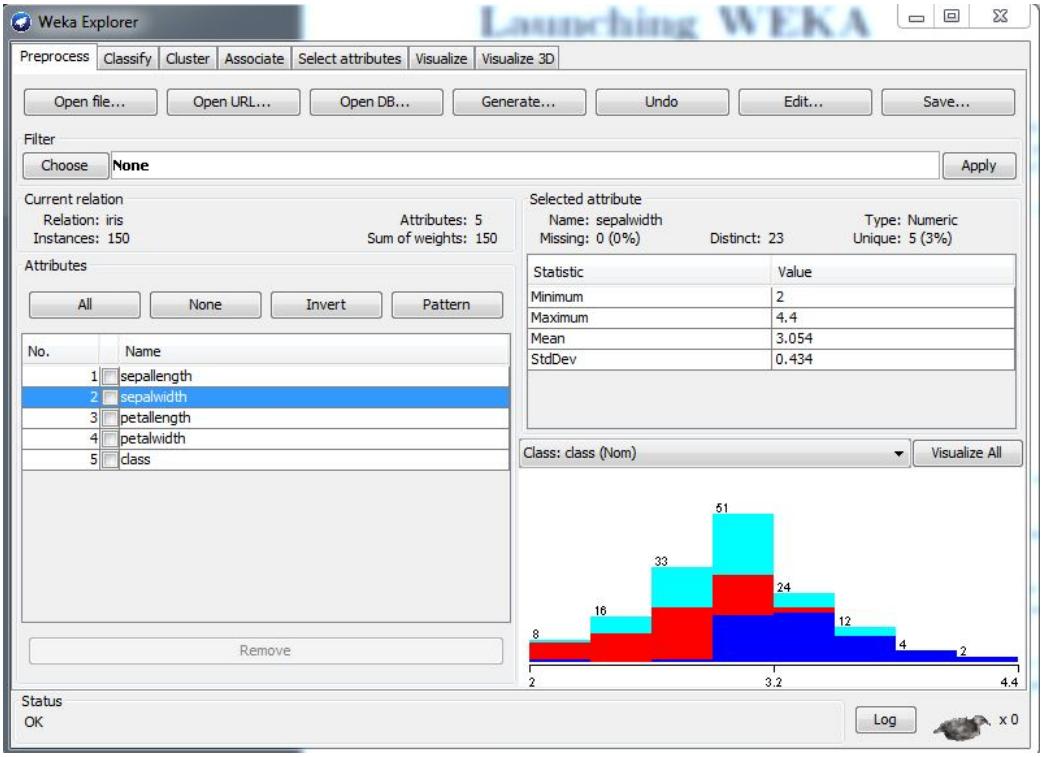
[20] Szczegółowe omówienie idei oraz zastosowania wszystkich algorytmów znajduje się w [84].

[21] Szczegółowe informacje o dostępnych metodach graficznej prezentacji danych zawiera dokumentacja techniczna dostępna pod adresem `http://docs.rapid-i.com/files/rapidminer/RapidMiner-5.2-Advanced-Charts-english-v1.0.pdf`.

wykresów i macierzy rozrzutu, bąbelkowych, map samoorganizujących się itp. Ponadto, jako jedyne z opisywanych rozwiązanie niekomercyjne, posiada możliwość graficznego przedstawienia struktury skupień w formie interaktywnego grafu (drzewiastego bądź radialnego). Użytkownik może kliknąć na wybrane skupienie by zobaczyć listę obiektów do niego należących.

Weka

Ostatni niekomercyjny program, omawiany w ramach niniejszego rozdziału, nosi nazwę Weka (ang. ***W**aikato **E**nvironment for **K**nowledge **A**nalysis*) i tworzony jest od 1997 w Nowej Zelandii. Weka to zbiór algorytmów uczenia maszynowego dedykowanych do realizacji zadań eksploracji danych. Algorytmy te mogą być stosowane bezpośrednio do zestawu danych lub za pomocą odpowiednich procedur wywoływanych z autorskiego kodu napisanego w języku Java. Weka zawiera narzędzia do wstępnego przetwarzania danych, klasyfikacji, zadań regresji, grupowania, generowania reguł asocjacyjnych i wizualizacji. Jednym z czynników wyróżniających to oprogramowanie od pozostałych jest fakt, iż posiada ono cztery różne interfejsy: *Explorer*, (do wydobywania wiedzy korzystając z szeregu zakładek), *Experimenter* (umożliwiający automatyczne wykonanie przygotowanych wcześniej eksperymentów), *KnowledgeFlow* (posiadający podobną funkcjonalność do trybu *Explorer* jednakże zrealizowany w oparciu o zasadę przeciągnij i upuść) oraz *Simple CLI* (udostępniający konsolę do przetwarzania wsadowego). W dalszej części tej sekcji zostanie omówiony tryb *Explorer* programu w wersji 3.7.10 beta, zaprezentowany na rysunku 3.14.



Rysunek 3.14: Interfejs w trybie *Explorer* programu Weka.
Źródło: Opracowanie własne

Interakcja użytkownika z programem odbywa się przez wybór odpowiedniej zakładki (z dostępnego zasobu siedmiu). Przykładowo pierwsza z nich służy wstępnemu przetwarzaniu i analizie eksploracyjnej, w formie statystyk opisowych jako średnia czy odchylenie standardowe i histogramów, dzięki czemu można oszacować rozkład danych, zidentyfikować wartości brakujące lub odstające i na tych przypadkach skupić dalszą analizę. W przypadku wybrania zakładki Cluster dedykowanej grupowaniu danych, można zastosować jeden z dziewięciu algorytmów analizy skupień m.in. jak k-średnich, hierarchiczny, EM, czy DBSCAN[22]. Jest to również jedyne rozwiązanie niekomercyjne, które posiada bezpośrednio zaimplementowany algorytm OPTICS. Niestety twórcy Weka nie zdecydowali się na dołożenie do niego procedury generującej skupienia (gdyż OPTICS domyślnie generuje jedynie specyficzne uporządkowanie obiektów, co zostało szczegółowo omówione w rozdziale 5). Ponadto, dla algorytmów gęstościowych dostępne są jedynie dwie miary podobieństwa takie jak odległość euklidesowa i miejska, zmodyfikowane tak, by możliwe było ich bezpośrednie zastosowanie do danych opisanych za pomocą atrybutów ilościowych i jakościowych. Wyniki procesu grupowania wizualizowane są wyłącznie jako macierz wykresów rozrzutu (lub pojedynczy trójwymiarowy wykres rozrzutu). Dlatego też to właśnie niewielkie możliwości wizualizacji rezultatów analizy skupień są największą wadą omawianego oprogramowania.

3.3 Podsumowanie

Celem niniejszego rozdziału było dokonanie przeglądu możliwości oprogramowania do wydobywania wiedzy z danych, w szczególności pod kątem zaimplementowanych algorytmów analizy skupień, ich graficznej reprezentacji oraz metod opisu danych. Porównanie dotyczyło zarówno programów komercyjnych jak również najbardziej popularnych rozwiązań udostępnianych za darmo (które w wielu przypadkach posiadają podobny zestaw funkcjonalności w odniesieniu do ich płatnych odpowiedników).

Wyniki przeglądu oprogramowania wskazują jednoznacznie, że żaden z testowanych pakietów analizy danych nie udostępnia interaktywnej metody graficznej reprezentacji skupień, dostosowanej do wizualizacji dużej liczby grup. Ponadto tylko niewielka liczba programów implementuje bardziej zaawansowane algorytmy analizy skupień (np. gęstościowe) i umożliwia ich zastosowanie do danych opisanych atrybutami ilościowymi oraz jakościowymi. Dlatego też uzasadniona wydaje się konieczność stworzenia autorskiego systemu wydobywania wiedzy z danych złożonych, który realizuje proces graficznej analizy eksploracyjnej (omówiony w rozdziale 6), poprzez reprezentację skupień (utworzonych przez zastosowanie algorytmu gęstościowego OPTICS) w formie map prostokątów, wykorzystującej cały dostępny obszar roboczy. Szczegóły dotyczące architektury i funkcjonowania takiego systemu znajdują się w rozdziale 7 niniejszej pracy.

[22] Szczegółowy opis zastosowanych algorytmów znajduje się w [88].

Rozdział 4

Metody opisu danych

Proces analizy danych zazwyczaj rozpoczyna się od etapów ich gromadzenia, opisu oraz czyszczenia[1]. Dane rzeczywiste najczęściej pochodzą z wielu heterogenicznych źródeł (jak bazy danych czy dzienniki zdarzeń), przez co mogą posiadać różną strukturę, szum informacyjny jak również brakujące wartości. Nawet w przypadku, gdy dane źródłowe są już wstępnie przetworzone i reprezentowane np. w postaci relacyjnej bazy danych, pozyskanie metadanych (rozumianych jako opisy kolumn bądź obowiązujące reguły biznesowe) jest kolejnym wyzwaniem. W wielu przypadkach metadane są słabo udokumentowane [14] i zachodzi potrzeba wielokrotnej komunikacji z ekspertem dziedzinowym by uzyskać stosowne wyjaśnienia. Dlatego też zanim będzie można zastosować techniki eksploracji danych (jak analizę skupień) w celu wykrycia potencjalnie interesujących związków i korelacji w danych, należy dokonać ich opisu, aby zrozumieć znaczenie czy potrzebę rejestrowania określonych parametrów. W literaturze przedmiotu [20, 14] funkcjonuje pojęcie eksploracyjnej analizy danych EDA (ang. *exploratory data analysis*, ***E**xploratory **D**ata **M**ining*) – jest to zbiór technik statystycznych, które mają na celu pomóc w zrozumieniu struktury danych źródłowych oraz ich charakterystyki. W zależności od problemu do rozwiązania, EDA może obejmować wszystko – od zastosowania prostych statystyk opisowych do wykorzystania modeli regresji lub wielowymiarowych technik eksploracyjnych [20].

Koncepcja eksploracyjnej analizy danych została stworzona przez amerykańskiego statystyka John'a Tukey'a w 1977 roku. EDA zostało zdefiniowane w [14] jako "wstępny proces odkrywania struktury zbioru danych za pomocą zestawień statystycznych, wizualizacji i innych środków". Celem tego procesu jest m.in.: maksymalizacja wiedzy o danych oraz ujawnienie ich wewnętrznej struktury, wykrycie anomalii i odchyleń, określenie założeń czy celów dalszej analizy. Silnie akcentowana jest również rola EDA jako pomocy w formułowaniu wstępnych hipotez odnośnie rozkładu danych czy korelacji między atrybutami, jak również wsparcie w wyborze najlepszych technik statystycznego opisu.

Celem niniejszego rozdziału jest przedstawienie wybranych technik eksploracyjnej analizy

[1] Przy założeniu, że analityk posiada już wiedzę dziedzinową oraz jasno sformułowany cel analiz.

danych wraz z ich klasyfikacją. Zostaną opisane zarówno proste miary tzw. centralnej tendencji (ang. *measures of central tendency*) jak średnia, mediana czy moda, ale również metody graficzne jak wykorzystanie histogramów czy wykresów częstości (ang. *frequency plots*). Należy jednak wyraźnie zaznaczyć, że bardzo istotne są typy danych z jakimi ma się do czynienia podczas procesu badawczego, gdyż od tego zależy jakie metody opisu danych możliwe są do zastosowania. Niektóre pozycje literaturowe jak [14], podają bardzo ogólną klasyfikację typów danych (atrybutów) na numeryczne, jakościowe i tekstowe. W [20] zastosowano podobny podział (na numeryczne, jakościowe i inne) jednakże został on odniesiony do typów danych występujących najczęściej w językach programowania (jak integer do reprezentowania liczb całkowitych, czy float do liczb rzeczywistych). Bardziej szczegółową klasyfikację przedstawiono w [19], gdzie wyróżniono dane dyskretne, ciągłe, nominalne, binarne, transakcyjne, symboliczne i szeregi czasowe. Ze względu na przedstawione rozbieżności w źródłach literaturowych, autor na potrzeby niniejszego rozdziału zakłada najbardziej ogólny podział danych na jakościowe i ilościowe.

4.1 Statystyka opisowa

Statystyka opisowa (ang. *descriptive statistics*) agreguje kilka różnych metod matematycznych mających na celu scharakteryzowanie dużej liczby danych, uwypuklając ich główne właściwości jak również określenie, które wartości powinny być traktowane jako izolowane. W związku z tym, istnieją dwa zasadnicze podejścia do statystycznego opisu danych: określenie parametrów liczbowych jak mediana czy moda lub bazowanie na graficznej reprezentacji (struktury i specyfiki) danych. Należy jednak nadmienić, że wspomniane podejście znacząco różni się od wnioskowania statystycznego (ang. *inferential statistics*). Statystyka opisowa ma na celu bardziej ilościowe lub graficzne podsumowanie (opisanie) zbioru danych, aniżeli dokonywanie predykcji czy wnioskowania odnośnie populacji, jaką ten zbiór miałby reprezentować. Zatem rezultaty opisywanego podejścia nie mogą zostać wprost uogólnione na inne, większe zestawy danych.

Do statystycznego opisu danych wykorzystywane są najczęściej dwa rodzaje miar: tendencji centralnej (ang. *central tendency*) oraz rozproszenia (ang. *dispertion*). Miary tendencji centralnej (położenia) takie jak średnia, mediana czy moda identyfikują typowe bądź "środkowe" wartości rozkładu danych, które w jakimś stopniu reprezentują cały zbiór. Intuicyjnie rzecz ujmując, pokazują pozycję, wokół której skupiają się wartości jakie przyjmuje dany atrybut. Oprócz oceny centralnej tendencji zbioru danych, analityk powinien również mieć pojęcie na temat ich rozproszenia. Najbardziej popularnymi miarami badającymi rozrzut danych są wariancja, odchylenie standardowe i rozstęp międzykwartylowy (ang. *interquartile range*).

Wykorzystanie tego typu miar pozwala wyznaczyć wartości odbiegające od normy, czy to z powodu błędów powstałych podczas gromadzenia danych, czy ze względu na naturalnie wstępujący szum informacyjny. Może to sygnalizować konieczność przeprowadzenia procesu oczyszczania lub normalizacji danych, jak również zwracać uwagę na potencjalnie interesujące przypadki, na których powinna skupić się dalsza eksploracja – przykładowo użytkownicy generujący bardzo duży ruch w sieci telekomunikacyjnej mogą być uznawani za interesujące

odchylenia. Należy tutaj nadmienić, że w procesie eksploracyjnej analizy danych, wymienione wcześniej miary powinny być traktowane jako wzajemne uzupełnienie, ponieważ każda z nich skupia się tylko na jednym aspekcie analizowanego zestawu danych. Dopiero w połączeniu dostarczają stosownej wiedzy odnośnie struktury danych. Przykładowo badając jednocześnie średnią (arytmetyczną) i medianę można oszacować ewentualną asymetryczność rozkładu [14].

Załóżmy, że w analizowanym zbiorze danych istnieje atrybut ilościowy X, oznaczający wiek klienta i niech $x_1, x_2, \ldots, x_M$ będzie zbiorem M wartości analizowanego atrybutu dla wszystkich obserwacji w zbiorze. Wówczas najbardziej popularna miara centralnej tendencji, czyli średnia arytmetyczna, jest zdefiniowana następująco:

$$\overline{x} = \frac{\sum_{i=1}^{M} x_i}{M} = \frac{x_1 + x_2 + \ldots + x_n}{M} \tag{4.1}$$

Mimo wysokiej popularności wynikającej z prostoty zastosowania tej miary, należy nadmienić, że jest ona bardzo wrażliwa na występowanie wartości izolowanych (skrajnie niepodobnych do pozostałych), co wpływa znacząco na zdolność do opisu danych. Przykładowo niech zbiór symbolizujący wiek dla wszystkich osób rejestrowanych w bazie jest następujący:

$$wiek = 25, 27, 23, 24, 25, 26, 25, 23, 26, 58.$$

Widać wyraźnie, że ostatnia wartość (58) w przedstawionej sekwencji znacząco odstaje od reszty. Średnia arytmetyczna uwzględniająca wszystkie wartości wynosi wówczas $28,2$. Jeżeli powtórzyć obliczenia, ale bez ostatniej wartości uznawanej za izolowaną, średnia wynosiłaby $\approx 24,9$, co jest lepszym uśrednieniem dla opisywanego przykładu biorąc pod uwagę rozkład pozostałych wartości. By uniknąć takich sytuacji, proponuje się wykorzystanie mediany jako miary centralnej tendencji [20]. Jednakże niektóre pozycje literaturowe [12, 35] sugerują również wykorzystanie tzw. średniej obciętej (ang. *trimmed mean*). Przy jej obliczaniu, elementy analizowanego zbioru (wartości) porządkuje się od najmniejszego do największego (lub odwrotnie), odrzuca się niewielki procent najbardziej ekstremalnych wartości na obu krańcach (na ogół równej liczności), a następnie oblicza się średnią arytmetyczną dla pozostałych elementów. Oczywiście należy uważać, by nie usunąć zbyt wielu elementów – zazwyczaj ogranicza się do maksimum 20% wszystkich wartości – ponieważ może to znacznie zaburzyć jakość otrzymanych wyników. Miara ta jest z powodzeniem używana do obliczania punktacji w wielu konkursach (np. skokach narciarskich), gdzie punkty przyznawane są przez kilku sędziów. Warto również podkreślić, że średnią arytmetyczną można bezpośrednio wyznaczyć tylko dla danych ilościowych. Istnieją również inne rodzaje średnich, jak geometryczna czy harmoniczna, jednakże nie będą one w niniejszej pracy szerzej opisywane m.in. ze względu na zależność od wartości, dla których są one obliczane. Jeżeli bowiem w analizowanym zbiorze występuje wartość zero, to zastosowanie wymienionych średnich mija się z celem, ponieważ wówczas średnia geometryczna zawsze wynosi 0, a harmoniczna wymaga dzielenia przez zero. Niemniej szczegółowe porównanie różnych rodzajów średnich znajduje się w [41, 35].

Dla rozkładów asymetrycznych, lepszą miarą tendencji centralnej jest mediana Md [12] – wartość, która dzieli analizowany zbiór[2] na dwa podzbiory o jednakowej liczności, przy założeniu, że wszystkie elementy są uprzednio uporządkowane (rosnąco lub malejąco). Mediana

[2]Pojęcie zbioru w niniejszej sekcji odnosi się do zestawu wartości przyjmowanych przez dany atrybut, dla wszystkich obiektów zapisanych w bazie danych.

jest zatem tzw. drugim kwartylem[3]. Załóżmy, że istnieje zbiór uporządkowany $x_1, x_2, \ldots, x_M$ elementów, dla którego należy wyznaczyć medianę. Jeżeli liczba elementów dla której wykonywane są obliczenia $M = 2t + 1$ jest nieparzysta, to mediana jest t-tą wartością w zbiorze. Jeśli natomiast $M = 2t$ jest parzyste, to jako medianę rozpatruje się wartości w przedziale $[t, t + 1]$ – nie musi ona zatem być unikalna. Dla wartości numerycznych, często definiuje się w takim przypadku medianę jako średnią arytmetyczną wartości między t oraz $t + 1$.

Mediana dla przytoczonego wcześniej zbioru *wiek* wynosi 25, co potwierdza jej zaletę jaką jest odporność na występowanie wartości odstających. Dodatkowo, jak zaznaczono w [20], mediana jest przydatna w sytuacji, gdy niektóre ekstremalne wartości w zbiorze danych nie są dostępne. Przykładowo bank mógłby zlecić dokonanie analizy centralnej tendencji na podstawie depozytów swoich klientów, ale najbogatsi z nich mogliby zastrzec prawo do udostępniania takich danych jakimkolwiek podmiotom. Wówczas wykorzystanie średniej byłoby niemożliwe (chyba, że do dostępnego ograniczonego zbioru), natomiast mediana jest możliwa do obliczenia (przy założeniu, że znana jest ogólna liczba elementów). Największą wadą mediany jest niemożność jej zastosowania do danych jakościowych, dla których nie można wyznaczyć relacji porządku. Przykładowo dla wartości atrybutu *kolor oczu* (jak piwne, zielone, czarne, niebieskie) nie da się określić ogólnie przyjętego uporządkowania, aby wyznaczyć medianę.

Najczęściej występującą wartość określa się mianem mody Mo (dominanty). Dominanta jest szczególnie użyteczna, ponieważ można ją wyznaczyć zarówno dla danych ilościowych i jakościowych. Dla przedstawionego wcześniej zbioru *wiek* moda wynosi 25, ponieważ jest to jedyna wartość, która występuje trzykrotnie (najwięcej razy). Niestety wadą tej miary jest możliwość wystąpienia więcej niż jednej wartości o tej samej liczbie wystąpień. Zbiory danych posiadające jedną, dwie lub trzy mody są kolejno nazywane unimodalnymi (ang. *unimodal*), bimodalnymi (ang. *bimodal*) oraz trójmodalnymi (ang. *trimodal*). Ogólnie rzecz biorąc, zestawy danych charakteryzujące się występowaniem dwóch lub więcej dominant nazywane są wielomodalnymi (ang. *multimodal*). Możliwa jest również sytuacja, w której każda analizowana wartość występuje tylko raz, wówczas przyjmuje się, że moda nie istnieje [12].

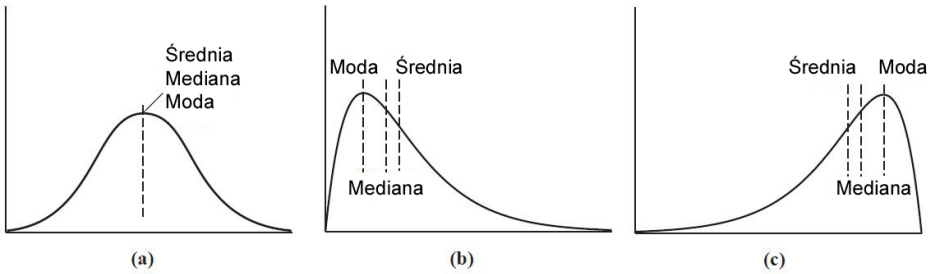
Jeżeli rozkład danych jest umiarkowanie asymetryczny oraz unimodalny można podać następującą zależność między dominantą, medianą i średnią:

$$Mo \approx 3 \cdot Md - 2 \cdot \overline{x} \tag{4.2}$$

gdzie Mo oznacza modę, Md medianę, natomiast $\overline{x}$ średnią arytmetyczną. Powyższa zależność implikuje, że znając wyłącznie wartości średniej i mediany można dokonać łatwej aproksymacji dominanty, dzięki czemu znacząco skraca się proces jej wyznaczenia. Dla przykładowego zbioru *wiek* tak wyznaczona wartość mody wynosi $3 \cdot 25 - 2 \cdot 28,2 \approx 75 - 56,4 \approx 18,6$. Obliczona wartość znacząco różni się od właściwej (25), ponieważ średnia arytmetyczna jest obciążona wartością 58, uznawaną za potencjalny błąd bądź szum informacyjny. Gdyby przeprowadzić obliczenia raz jeszcze, tym razem wykorzystując średnią obciętą $(24, 9)$, to wówczas aproksymowana moda wynosiłaby $25,2$ co jest znacząco bliższe oczekiwaniom.

Jeżeli zbiór danych jest unimodalny oraz występuje symetryczność rozkładu, to mediana, średnia i moda wskazują dokładnie tą samą wartość jak zobrazowano na rysunku 4.1a. Jed-

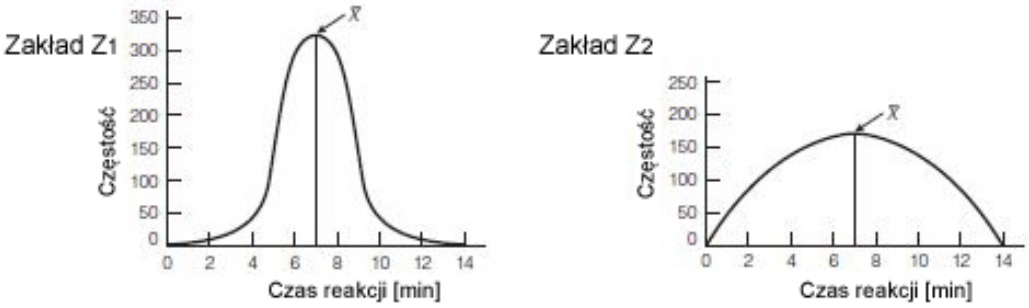
[3]Pojęcie kwartylu zostanie wyjaśnione w dalszej części niniejszego rozdziału.



Rysunek 4.1: Wpływ rozkładu danych na miary centralnej tendencji.
Źródło: [12]

nakże rzeczywiste zestawy danych rzadko kiedy posiadają takie właściwości. Rozkład może być prawostronnie skośny (ang. *positively skrewed*), wówczas moda ma wartość mniejszą niż mediana (rysunek 4.1b) lub lewostronnie skośny (ang. *negatively skrewed*), co warunkuje, że dominanta jest większa od mediany, jak pokazano na rysunku 4.1c.

W literaturze przedmiotu [22] często wymienia się również miary takie jak średnia z wartości skrajnych (ang. *midrange*), minimum i maksimum. Ich popularność warunkowana jest niską złożonością obliczeniową, co jest istotne w kontekście analizy dużych zbiorów danych złożonych. Wśród wad należy zaakcentować wrażliwość na występowanie wartości izolowanych, jak również ograniczenie tylko do skali porządkowej (ang. *ordinal scale*).



Rysunek 4.2: Ocena rozpiętości danych na podstawie czasu reakcji zakładów świadczących usługi medyczne.
Źródło: [12]

Kolejne podejście do analizy danych jest skoncentrowane na ocenie rozpiętości danych wokół średniej tj. na zmierzeniu całkowitej odległości między każdym elementem zbioru a wartością uznawaną za środkową (typową). Są to tzw. miary rozproszenia, które opisują zróżnicowanie bądź heterogeniczność rozkładu próbek (analizowanych wartości). Uwzględniają one wykorzystanie m.in. kwartyli, percentyli czy rozstępu międzykwartylowego. Znaczenie tych miar zostanie zaprezentowane na następującym przykładzie. Załóżmy, że pewna firma zastanawia się nad zakontraktowaniem usług medycznych i w okolicy są dwa zakłady Z_1 oraz Z_2, które je udzielają. Po dokonaniu wywiadu środowiskowego okazuje się, że oba zakłady charakteryzują się bardzo podobnym średnim czasem reakcji (przyjazdu karetki) na poziomie odpowiednio

7,4 i 7,6 minut. Dodatkowo niech dominanta i mediana będą tożsame ze średnią. W takim przypadku niemożliwe jest podjęcie jednoznacznej decyzji bazując tylko na miarach centralnej tendencji. Dlatego też, aby ocenić rozproszenie danych, skonstruowano dwa wykresy na rysunku 4.2, które pokazują rozkład czasu reakcji dla każdego z dwóch zakładów. Na przedstawionym rysunku widać wyraźnie, że wykres zakładu drugiego jest dużo płytszy, niż ten dla pierwszego. Dzieje się tak, ponieważ wartości powiązane z zakładem numer dwa są dużo bardziej rozproszone niż konkurencji. Innymi słowy, zakład Z_2 charakteryzuje duża zmienność czasu reakcji. Zakład Z_1 posiada bardziej spójny czas reakcji, a poszczególne zarejestrowane wartości są zgrupowane zdecydowanie bliżej średniej niż miało to miejsce dla Z_2. Zatem z przedstawionej prostej analizy wynika, że lepszym wyborem byłby zakład pierwszy.

Najpopularniejszą miarą rozrzutu (dla zbioru $x_1, x_2, \ldots, x_M$) elementów jest wariancja zdefiniowana jako:

$$\sigma^2 = \frac{1}{M}\sum_{i=1}^{M}(x_i - \overline{x})^2 \tag{4.3}$$

Jest to zatem średnia kwadratów różnic między wartością średnią a pojedynczymi wartościami danych. Należy jednak nadmienić, że w wielu pozycjach literaturowych jak również bibliotekach statystycznych spotyka się nieco inną definicję wariancji:

$$\sigma^2 = \frac{1}{M-1}\sum_{i=1}^{M}(x_i - \overline{x})^2 \tag{4.4}$$

Taka zmiana zapewnia uzyskanie nieobciążonego szacowania. Jednakże dla dużej liczby danych, różnica między wartościami uzyskanymi przez zastosowanie wzoru 4.3 oraz 4.4 jest pomijalna [20].

Często wygodniej korzystać jest z odchylenia standardowego (ze względu na zachowanie skali pomiaru tożsamej z bazowym zestawem danych), czyli pierwiastka kwadratowego z wariancji, zdefiniowanego wzorem:

$$\sigma = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(x_i - \overline{x})^2} \tag{4.5}$$

Odchylenie standardowe jest używane w statystyce opisowej do zdefiniowania przedziałów, które zawierają większość analizowanych elementów (wartości). Jeżeli rozkład danych jest symetryczny, zdecydowana większość wartości z analizowanego zbioru (około 95% z nich) znajduje się w przedziale określonym przez $\overline{x} \pm 2 \cdot \sigma$. Jest to tzw. 95% przedział ufności. Wartości znajdujące się poza wyznaczonym przedziałem ufności, można traktować jako szum informacyjny. Ogólnie rzecz biorąc, używając nierówności Czebyszewa można udowodnić, że co najmniej $(1 - \frac{1}{t^2}) \cdot 100\%$ wartości jest oddalonych maksymalnie o t odchyleń standardowych od średniej [22].

Kolejną, prostą miarą rozrzutu jest rozstęp (ang. *range of values*). Miara rozstępu R dla zbioru elementów $x_1, x_2, \ldots, x_M$ zadana jest jako:

$$R = \max_i(x_i) - \min_i(x_i), \quad i = 1, \ldots, M. \tag{4.6}$$

Tabela 4.1: Główne parametry rozrzutu danych

Przypadek	Wariancja	Odchylenie standardowe	Rozstęp
Przypadek 1	100,16	≈10,01	35
Przypadek 2	≈1,65	≈1,28	4

Rozstęp jest zatem różnicą między wartością maksymalną i minimalną w analizowanym zbiorze. Niestety miara ta, mimo intuicyjnej koncepcji, jest bardzo czuła na występowanie wartości izolowanych, przez co jest przydatna tylko w celu uzyskania bardzo ogólnego pojęcia na temat rozproszenia, badając jednocześnie wiele rozkładów (np. dla wielu atrybutów naraz). Ponadto jak zaznaczono w [35] rozstęp jest niewrażliwy na kształt rozkładu danych, gdyż zależy tylko od dwóch wartości skrajnych.

Wyliczenia wybranych miar rozrzutu, dla przykładowego zbioru *wiek*, przedstawiono w tabeli 4.1. Obliczenia wykonano zarówno uwzględniając wartość izolowaną 58 (co oznaczono jako przypadek pierwszy), jak również wykluczając tę wartość z obliczeń (w drugim przypadku). Analizując wyniki zawarte w tabeli 4.1, widać wyraźnie negatywny wpływ występowania wartości izolowanej na wszystkie parametry rozrzutu danych.

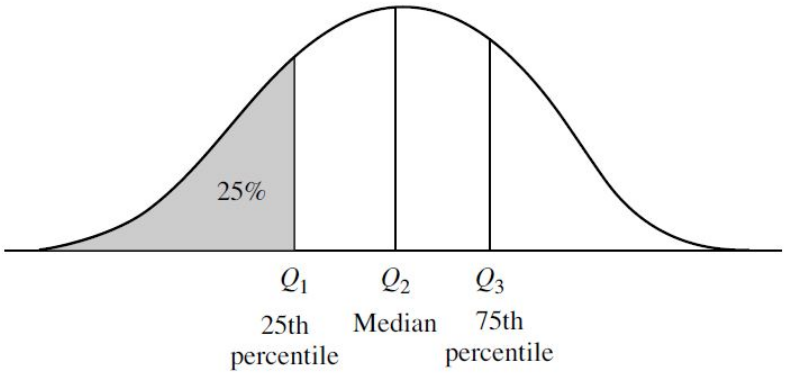
Zanim zostanie omówiona kolejna miara rozproszenia, czyli rozstęp międzykwartylowy, niezbędnym jest wprowadzenie kilku pojęć z tym związanych, wywodzących się ze statystyki, takich jak kwantyl, percentyl, czy kwartyl. Niech $x_1, x_2, \ldots, x_M$ będzie posortowanym rosnąco zbiorem obserwacji (wartości) przyjmowanych przez pewien atrybut numeryczny X. Kwantylem (ang. *quantile*) rzędu α, $0 < \alpha < 1$ cechy X nazywamy liczbę q_α taką, że $\alpha \cdot 100\%$ elementów zbioru $x_1, x_2, \ldots, x_M$ ma wartość badanej cechy nie większą od q_α. Z punktu widzenia analizy danych, wyznaczenie kwantyli obejmuje proces podziału uporządkowanego zbioru na $\frac{1}{\alpha}$ równolicznych[4] podzbiorów w ten sposób, że kwantyle są wartościami wyznaczającymi granice między tymi grupami. Kwantyl rzędu 0,5 jest zazwyczaj tożsamy z medianą (w przypadku nieparzystej liczby analizowanych elementów)[5], gdyż jest to punkt dzielący rozkład danych na dwie połowy. W praktyce jednak, do opisu danych wykorzystywane są pojęcia takie jak percentyle (ang. *percentiles*), decyle (ang. *deciles*) czy kwartyle (ang. *quartiles*), będącymi szczególnymi przypadkami kwantyli.

Percentyl jest kwartylem rzędu $\frac{i}{100}$, gdzie $i = 1, 2, \ldots, 99$. Zatem reprezentuje dowolną z 99 wartości, które dzielą zbiór badanych elementów na 100 podzbiorów tej samej wielkości. Przykładowo dziesiąty percentyl oznacza, że 10% elementów w uporządkowanym zbiorze ma wartość mniejszą lub równą wartości tego percentyla, natomiast 90% elementów ma wartość mu równą lub większą.

Podobnie można zdefiniować kwartyle – są to kwantyle rzędu $\frac{1}{4}$, $\frac{2}{4}$ lub $\frac{3}{4}$ oznaczane kolejno jako Q_1, Q_2 oraz Q_3. Wobec tego dla pierwszego kwartyla Q_1 (zwanego również dolnym) istnieje 25% danych przed nim i 75% za nim, natomiast dla trzeciego kwartyla Q_3 (nazywanego górnym) sytuacja jest odwrotna. Kwartyl drugi Q_2 dzieli analizowany zbiór na dwie połowy,

[4]Należy jednak zaznaczyć, że w analizowanym zbiorze wartości, może nie być takiej, dzięki której podział będzie zawsze równoliczny.

[5]W przypadku parzystej liczby elementów branych pod uwagę przy wyznaczaniu mediany, jest to zależne od jej definicji, ponieważ często wykorzystuje się średnią arytmetyczną.



Rysunek 4.3: Nazewnictwo szczególnych przypadków kwantyli.
Źródło: [22]

przez co jest utożsamiany z medianą. Opisane zależności zostały przedstawione graficznie na rysunku 4.3.

Znając definicję kwartyli można przystąpić do określenia miary rozstępu międzykwartylowego (ćwiartkowego) IQR. Jest to miara rozrzutu zmiennej, podobna do odchylenia standardowego, jednak bardziej odporna na wartości odstające. Formalnie IQR jest zdefiniowany jako:

$$IQR = Q_3 - Q_1 \tag{4.7}$$

Rozstęp międzykwartylowy można interpretować jako różnicę między trzecim a pierwszym kwartylem. Rozstęp ćwiartkowy zasadniczo wyodrębnia środkowe 50% danych i podobnie jak przytoczona wcześniej miara rozstępu R, bazuje tylko na dwóch wielkościach. W przeciwieństwie do poprzedniej miary, IQR nie dotyczy jednak problem wrażliwości na występowanie wartości izolowanych (chyba, że ponad połowa analizowanych elementów może być zaklasyfikowana jako szum informacyjny). Wartości kwartyli[6] oraz rozstępu ćwiartkowego dla przykładowego zbioru *wiek* zostały zaprezentowane w tabeli 4.2.

Statystyka opisowa nie zawsze jest jednak wystarczająca do skutecznej analizy danych. Przykładowo w zadaniu wykrywania odchyleń, dużo lepiej sprawdzają się metody graficzne jak generowanie histogramu, co pozwala bardzo szybko wychwycić pojedyncze, odstające wartości. Dlatego też w dalszej części tego rozdziału zostaną przeanalizowane graficzne techniki opisu danych wraz z uwzględnieniem ich wad i zalet.

Tabela 4.2: Rozrzut bazujący na kwartylach

Kwartyl pierwszy Q_1	Kwartyl drugi Q_2	Kwartyl trzeci Q_3	Rozstęp ćwiartkowy IQR
24	25	26	2

[6]Szczegółowe informacje odnośnie wyznaczania poszczególnych kwartyli zawarto w [41].

4.2 Metody graficzne

Podstawowym sposobem graficznego przedstawienia rozkładu danych jest zastosowanie histogramu. Histogram dla atrybutu ilościowego X dzieli rozkład empiryczny cechy na rozłączne podzbiory zwane kubłami (ang. *bins*). Zazwyczaj szerokość każdego kubła jest jednakowa[7]. Dany kubeł reprezentowany jest przez prostokąt, którego wysokość jest równa liczbie lub częstości względnej[8] wartości należących do reprezentowanego podzbioru. Jeżeli X jest atrybutem jakościowym, prostokąty rysowane są dla każdej wartości tego atrybutu, a ich wysokość wskazuje na częstość występowania danej wartości w analizowanym zbiorze. Wówczas jednak źródła literaturowe [12], tak sporządzony wykres, określają mianem wykresu słupkowego (analizy częstości), natomiast termin histogram preferowany jest w przypadku danych ilościowych.

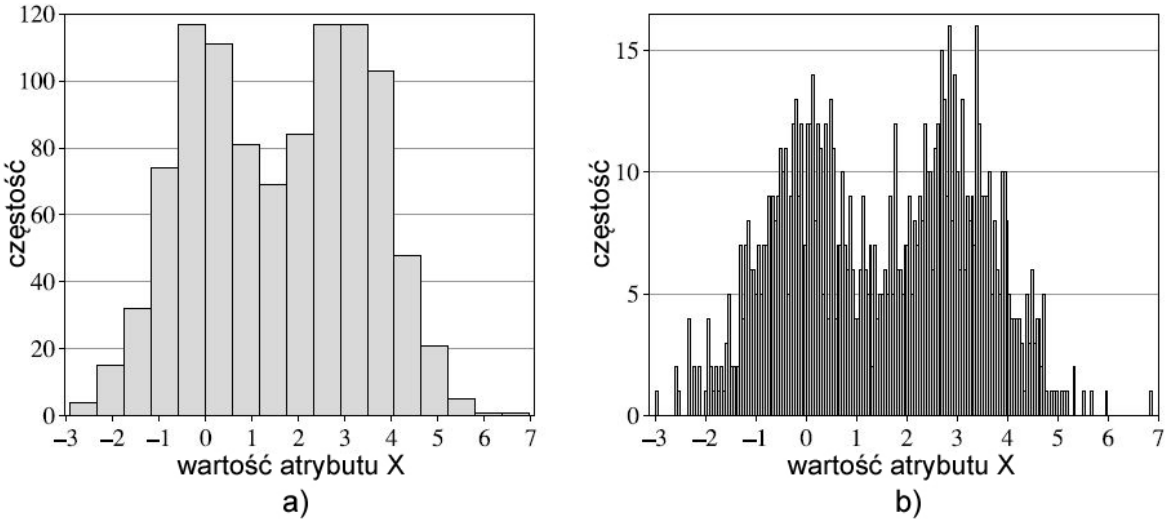
Generalnie wyróżnia się dwa rodzaje histogramów: o jednakowej szerokości przedziałów w dziedzine wartości atrybutu (ang. *equi-spaced*) oraz o możliwie jednakowej liczbie wystąpień (częstości) wartości atrybutu w podzbiorach (ang. *equi-depth*). W przypadku histogramów pierwszego typu (*equi-spaced*), zbiór przyjmowanych wartości przez analizowany atrybut jest dzielony na przedziały o jednakowej wielkości. Zaletą tego schematu postępowania jest możliwość, by przedziały zostały zdefiniowane a priori. Konstrukcja takich histogramów wymaga tylko jednokrotnego przeglądu danych, przez co jest relatywnie szybka. Ponadto umożliwia to łatwe porównanie histogramów wyznaczonych dla różnych cech. W przypadku podejścia alternatywnego (ang. *equi-depth*), dane zostają podzielone w ten sposób, że we wszystkich przedziałach znajduje się ta sama liczba wartości. Prowadzi to do powstania histogramu, gdzie krańce przedziałów to kwintyle [14]. Histogramy tego typu wykorzystane są m. in. do szacowania kosztu wykonania zapytania SQL w bazach danych.

Dla małych zbiorów (wartości) histogramy mogą prowadzić do mylnych wniosków: "przykładowe wahania wartości lub alternatywne wybory końców przedziałów mogą doprowadzić do bardzo różnych diagramów. Przy różnych wyborach przedziałów lub różnych małych próbkach może pojawić się, a potem zniknąć jawna wielomodalność. Kiedy rozmiar zbioru danych wzrasta, szansa wystąpienia tego rodzaju zjawisk maleje" [23]. Dobór wielkości przedziałów jest zatem sprawą dość istotną, gdyż znacząco wpływa na wygląd histogramu i ewentualne wnioski z analizy. Przykład histogramów wyznaczonych na podstawie atrybutu numerycznego X przyjmującego wartości rzeczywiste z zakresu od -3 do 7 i tysiąca obiektów, prezentuje rysunek 4.4. Przytoczony zestaw danych został wygenerowany przez połączenie dwóch zbiorów danych o rozkładach normalnych ze średnią w punktach 0 i 3 oraz odchyleniem standardowym równym jeden. Na rysunku 4.4b zaznaczono sytuację, gdzie źle dobrano liczbę przedziałów, w stosunku do właściwej (rys. 4.4a). Wybór zbyt dużej liczby przedziałów prowadzi zazwyczaj do powstania bardzo niejednorodnego i rozproszonego histogramu, na podstawie którego trudno jest odróżnić prawdziwe punkty szczytowe (wywodzące się z rozkładu danych) od sztucznych.

Niestety nie istnieją jasne wytyczne, pozwalające wybrać optymalną liczbę przedziałów dla każdego rodzaju danych, jednakże można podać pewne rekomendacje. Reguła Struges'a [8]

[7] W przypadku różnych szerokości należy dokonać korekty częstości biorąc pod uwagę najmniejszy przedział. Szczegóły tej procedury podane są w [29].

[8] Poprzez częstość względną rozumie się stosunek liczby wartości należących do danego przedziału histogramu do całkowitej liczby wartości. Jest to wykorzystywane do normalizacji danych na histogramie.



Rysunek 4.4: Wpływ doboru szerokości przedziału na wygląd i interpretacje histogramów. Źródło: [8]

proponuje, by wybrać liczbę *nbn* przedziałów zgodnie z formułą:

$$nbn = \lceil \log_2(M) + 1 \rceil, \tag{4.8}$$

gdzie M to liczba próbek (wartości). Jeżeli przedziały danych byłyby jednakowej wielkości h, ich liczba może zostać ustalona na podstawie parametru h zgodnie ze wzorem:

$$nbn = \left\lceil \frac{\max_i(x_i) - \min_i(x_i)}{h} \right\rceil, \tag{4.9}$$

gdzie $x_1, \ldots, x_M$ to zbiór wartości poddawany wizualizacji. Wówczas na podstawie [8] rozpiętość przedziału h można dobrać jako:

$$h = \frac{3,5 \cdot \sigma}{M^{\frac{1}{3}}}, \tag{4.10}$$

gdzie σ to odchylenie standardowe. Jest to tzw. reguła Scott'a. Daje ona dobre wyniki w przypadku danych o rozkładzie normalnym [57]. W pracy [8] podano również inną metodę doboru parametru h jako:

$$h = \frac{2 \cdot IQR}{M^{\frac{1}{3}}}, \tag{4.11}$$

gdzie IQR to rozstęp międzykwartylowy. Niniejsza formuła nosi nazwę reguły Freedman–Diaconis. Rysunek 4.4a przedstawia histogram podzielony na 17 przedziałów, ustalonych zgodnie z propozycją Freedman'a–Diaconis'a. Wykorzystanie rozstępu ćwiartkowego zamiast odchylenia standardowego skutkuje nieco mniejszą podatnością na występowanie wartości izolowanych w analizowanym zbiorze (wartości)[9].

Niestety wszystkie opisane metody wyznaczania liczby przedziałów celem wykonania histogramu, są podatne na występowanie szumu informacyjnego, gdyż dokonują podziału całego

[9]Inne sposoby wyznaczania liczby i wielkości przedziałów zostały opisane m.in. w [75].

zbioru wartości analizowanego atrybutu na kubły jednakowych rozmiarów. Nawet pojedyncza wartość izolowana może spowodować, że zakres ten (między najmniejszą a największą wartością) będzie relatywnie duży, przez co dla małej liczby przedziałów, stają się one bardzo duże (szerokie), agregując wiele elementów. W przypadku dużej liczby przedziałów, wiele z nich może być pustych. Żeby uniknąć takiej sytuacji można usunąć określony procent wartości skrajnych z analizowanego przedziału (podobnie jak to miało miejsce przy wyznaczaniu średniej obciętej) lub dokonać podziału na nierównomiernie rozłożone przedziały. Wykorzystanie przedziałów o różnej wielkości jest jednak niekorzystne, jeżeli analityk chciałby porównać kilka różnych histogramów ze sobą. Należy również nadmienić, że dla dużej liczby danych ilościowych tego typu wizualizacje przestają być czytelne lub wręcz komplikują zrozumienie zbioru. Dlatego też przed wygenerowaniem histogramu najczęściej wykonuje się dyskretyzację danych, tam gdzie jest to możliwe i wskazane. W przypadku analizy rzeczywistych zbiorów danych złożonych, dyskretyzacja musi czasem być przeprowadzona przez eksperta dziedzinowego. Taka sytuacja miała miejsce dla zbioru *cell_loss* i atrybutu *strata*, ponieważ tylko ekspert mógł określić na ile i na jakich poziomach rozróżniać niedostępność urządzenia nadawczo-odbiorczego.

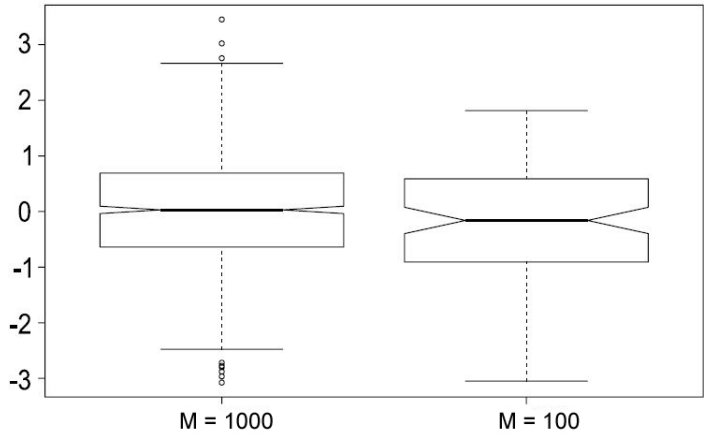
Dobrą alternatywą dla histogramów jest wykorzystanie tzw. wykresów pudełkowych (ang. *boxplots*) do wizualizacji głównych charakterystyk zbioru wartości numerycznych. Pozwalają one ująć na jednym diagramie informacje odnośnie centralnej tendencji, rozproszenia i kształtu rozkładu (empirycznego) badanej cechy. Wykres pudełkowy (skrzynkowy) jest podobny do histogramu w tym sensie, że umożliwia wizualną ocenę rozkładu danych, natomiast prezentacja tych informacji odbywa się na niższym poziomie szczegółowości, dzięki czemu wykresy skrzynkowe zajmują znacznie mniej miejsca, przez co łatwiej jest porównać rozkłady wielu cech jednocześnie.

Proces tworzenia wykresu skrzynkowego rozpoczyna się od zaznaczenia prostokątnego obszaru (zwanego pudełkiem), sięgającego od pierwszego do trzeciego kwartylu. Samo pudełko odwzorowuje zatem rozstęp międzykwartylowy. Następnie ciągłą linią, znajdującą się wewnątrz pudełka, zaznaczana jest mediana zestawu danych. Dodatkowo tworzone są pionowe[10] linie (zwane wąsami), które reprezentują określony zakres wartości – zwyczajowo wąsy rozciągają się od najmniejszej do największej wartości w zbiorze (często pomijając te klasyfikowane jako szum informacyjny). Krańce wąsów mogą jednak reprezentować alternatywne wartości jak:

- najniższą i największą wartość zbioru znajdującą się w obrębie $1,5 \cdot IQR$ względem mediany [60],
- jedno odchylenie standardowe poniżej i powyżej średniej [25],
- drugi i 98 percentyl [51].

Wartości izolowane (czyli znajdujące się poza wąsami) na wykresie zaznaczane są jako kropki bądź małe okręgi. Przykład dwóch wykresów pudełkowych (dla danych o rozkładzie normalnym ze średnią zero i odchyleniem standardowym równym jeden) zaprezentowano na rysunku 4.5. Wykres po stronie lewej został stworzony wykorzystując 1000 wartości, natomiast ten po stronie prawej wizualizuje ich jedynie sto.

[10] Przy założeniu, że rozstęp międzykwartylowy też został zaznaczony na osi rzędnych (pionowej).



Rysunek 4.5: Przykład wykresów pudełkowych dla próbek różnych rozmiarów.
Źródło: [8]

Wykresy na rysunku 4.5 posiadają dodatkowo wcięcia (ang. *notches*) wewnątrz prostokątnego pudełka. Za ich pomocą określany jest 95% przedział ufności dla mediany. Same pudełko w dalszym ciągu odpowiada rozstępowi ćwiartkowemu obejmującym 50% środkowej części zbioru wartości. Maksymalna długość każdego wąsa jest równa półtorej wielkości rozstępu międzykwartylowego. Jednakże jeśli nie istnieje w analizowanym zbiorze wartość odpowiadająca maksymalnej długości wąsa, jest on skracany do napotkania kolejnego elementu zbioru. Wartości znajdujące się poza wąsami klasyfikowane są jako szum informacyjny i zaznaczane małymi okręgami.

Analizując rysunek 4.5 można stwierdzić co następuje:

- mimo, że oba wykresy zostały stworzone na podstawie tego samego zbioru danych, wyglądają inaczej, ponieważ użyto próbek różnej wielkości (odpowiednio tysiąc elementów dla lewego i sto dla prawego wykresu),
- wcięcie na pierwszym wykresie pudełkowym (po lewej stronie), reprezentujące 95% przedział ufności dla mediany, jest dużo mniejsze niż wcięcie na prawym wykresie, ze względu na większą liczbę elementów danych do stworzenia lewego wykresu,
- długości wąsów na obu wykresach są diametralnie różne, pomimo takiego samego rozkładu, gdyż dla mniejszej liczby elementów maksymalna wartość nie była większa niż 2, natomiast minimum oscylowało wokół -3,
- w przeciwieństwie do wykresu znajdującego się po stronie lewej rysunku, wykres prawy nie posiada żadnych wartości izolowanych. Jest to znowu spowodowane mniejszą liczbą elementów. Jak podano w [8] prawdopodobieństwo wystąpienia wartości izolowanej w próbce 100 elementów o rozkładzie normalnym wynosi zaledwie $0,7\%$.

Specyficznym wariantem wykresów pudełkowych, są tzw. wykresy diamentowe (ang. *diamond plot*). Tego typu wykres ukazuje rozkład danych wykorzystując jego średnią i odchylenie

standardowe, podczas gdy wykresy pudełkowe operują na pojęciu mediany oraz kwartyli. Dane tego typu (średnią i odchylenie standardowe) również można zaznaczać na wykresie pudełkowym, uzyskując interesującą hybrydę [75].

Literatura związana z eksploracyjną analizą danych wyszczególnia również dwa podejścia do wizualizacji, bazujące na notacji kwantyli: wykres kwantyli (ang. *quantile plot*) oraz kwantyl–kwantyl (ang. *quantile—quantile plot*). Pierwszy typ wykresu wizualizuje wszystkie badane wartości z uporządkowanego (rosnąco) zbioru $x_1, x_2, \ldots, x_M$. Dla każdego elementu x_i wyznaczana jest wartość procentowa fp_i, która określa, że fp_i procent danych znajduje się przed wartością x_i. Przykładowo dla uporządkowanego zbioru wiek:

$$wiek_{upr} = 23, 23, 24, 25, 25, 25, 26, 26, 27, 58$$

element 24 (będący jednocześnie pierwszym kwartylem) zostanie powiązany z wartością procentową $0,25$. Świadczy to o tym, że 25% zbioru danych znajduje się przed tym elementem. Wykres kwantyli jest zatem wykresem zależności między x_i oraz fp_i [22]. Wykres kwantyl–kwantyl, jak nazwa wskazuje, jest zatem odwzorowaniem kwantyli dwóch różnych obserwacji (zbiorów wartości) względem siebie. Niestety tego typu diagramy stają się nieczytelne, jeżeli wizualizowany zbiór zawiera bardzo dużo wartości, dlatego też nie zostaną wykorzystane przy analizie cech rzeczywistych, złożonych zbiorów danych i nie będą szerzej opisywane[11]. Pozostałe graficzne techniki analizy eksploracyjnej spotykane w literaturze przedmiotu jak wykresy rozrzutu czy metody pikselowe, zostały przez autora niniejszej pracy opisane w [49].

Współcześnie statystyka opisowa, nawet wspomagana przez graficzne metody prezentacji, nie jest wystarczającym narzędziem badawczym. Celem nowoczesnych, inteligentnych systemów analizy jest wykorzystanie ukrytej wiedzy w danych. Wiedza ta może mieć formę pewnych ukrytych wzorców np. korelacji między atrybutami danych. Dlatego też coraz częściej do tego typu systemów integruje się techniki eksploracji jak analiza skupień, celem odkrycia nowej, potencjalnie użytecznej w dalszym procesie badawczym, wiedzy. Zastosowanie algorytmów analizy skupień pozwala relatywnie szybko zapoznać się ze strukturą analizowanego zbioru (ponieważ dowiadujemy się ile występuje grup obiektów oraz jak liczne one są), jak również zapewnia, że obiekty w danej grupie są do siebie maksymalnie podobne, zatem są ze sobą skorelowane ze względu na pewne cechy (uwzględniane przy grupowaniu).

4.3 Reprezentacja opisowo–eksploracyjna skupień

Metody analizy eksploracyjnej można zastosować zarówno przed dokonaniem właściwego procesu odkrywania wiedzy, by zapoznać się wstępnie ze specyfiką zbioru danych wejściowych, jak również podczas realizowania konkretnego zadania eksploracji (wykorzystując np. analizę skupień) do opisu rezultatów i łatwiejszej interpretacji uzyskanych wzorców. Autor niniejszej pracy zdecydował się wybrać gęstościowe algorytmy grupowania (opisane szczegółowo w następnym rozdziale) jako podstawowe narzędzie do odkrywania nowych, potencjalnie użytecznych wzorców. Dla utworzonych skupień przewiduje się budowę odpowiednich i prostych w analizie reprezentantów. W pracy [68] zaproponowano oraz wstępnie przetestowano

[11] Bardziej szczegółowe informacje o wykresach opartych o kwantyle znajdują się m.in. w [51] oraz [12].

eksperymentalnie (pod kątem przeszukiwania), cztery[12] koncepcje tworzenia reprezentantów skupień:

- jako część wspólną wśród deskryptorów opisujących obiekty należące do jednej grupy (używając operatora koniunkcji logicznej AND),
- jako unikalny zbiór deskryptorów zawartych w opisach obiektów jednej grupy,
- jako siedem najczęściej występujących deskryptorów wśród obiektów danej grupy,
- jako siedem najrzadziej występujących deskryptorów w ramach jednej grupy.

Zaletą pierwszej proponowanej koncepcji jest fakt, że można w łatwy sposób dostrzec jakie cechy są wspólne dla wszystkich obiektów danej grupy oraz porównać ją na tle występowania unikalnych cech z innymi skupieniami. Ponadto opis reprezentanta tego typu jest dość zwięzły i nigdy nie przekroczy liczby atrybutów opisujących obiekty.

Druga proponowana koncepcja opisu grup daje pełniejszy obraz na temat zawartości danej grupy. Niestety okazuje się ona także dużo trudniejsza w interpretacji, szczególnie jeżeli w obrębie skupienia występuje dużo unikalnych deskryptorów.

Następne sposoby opisu skupień, trzeci i czwarty, są do siebie bardzo podobne oraz pozwalają użytkownikowi na kontrolę długości reprezentanta – wartość siedmiu deskryptorów została dobrana arbitralnie (by uniknąć zbyt dużego stopnia skomplikowania), ale może to być dowolna wartość zależna od celu eksploracji czy rozkładu danych wejściowych. Podejście wykorzystujące najczęściej występujące pary atrybut–wartość promuje grupy spójne, natomiast podejście oparte na cechach najrzadziej występujących akcentuje przede wszystkim unikalne charakterystyki w ramach danego skupienia.

Po przeprowadzeniu wstępnych eksperymentów, opisanych w [68] oraz [47], dotyczących skuteczności metod reprezentacji skupień (w zadaniu przeszukiwania i interpretacji takiej struktury), zdecydowano się wybrać do dalszych badań koncepcję wykorzystującą operator koniunkcji logicznej AND, wykorzystywany szeroko ze względu na swoją intuicyjność przez wyszukiwarki internetowe. Przykładowo dla zbioru danych telekomunikacyjnych odnośnie funkcjonowania 143486 urządzeń nadawczo-odbiorczych (po zastosowaniu gęstościowego algorytmu analizy skupień) otrzymano 7933 grup, a reprezentant wybranego skupienia zgodny z opisywaną koncepcją kształtuje się następująco: *(czyProblem, 1)* ***AND*** *(obszarId, 41)* ***AND*** *(dostawcaId, 4)* ***AND*** *(data, 2010-06-23)* ***AND*** *(czyWorkflow, 1)* ***AND*** *(czyWoINN, 1)* ***AND*** *(czyWoTeren, 0)* ***AND*** *(czestosc, 2)* ***AND*** *(czasTrwania, 1)* ***AND*** *(czyPlanowane, 0)*. Skupienie to symbolizuje urządzenia znajdujące się w regionie 41, pochodzące od dostawcy 4, które są zakwalifikowane do zbadania przez dział utrzymania sieci, ponieważ dwukrotnie nie były dostępne dnia 23 czerwca 2010. Koncepcja z wykorzystaniem spójnika logicznego OR została odrzucona, ponieważ reprezentanci większych grup posiadali zbyt długi opis deskryptorowy – przykładowo analiza (przez analityka) reprezentanta składającego się z 146 unikalnych deskryptorów i jego porównanie z innymi, jest zadaniem czasochłonnym i często niemożliwym. W przypadku koncepcji

[12] Należy również nadmienić, że w pracy [47] odniesiono się do spotykanych w literaturze przedmiotu innych, popularnych koncepcji reprezentacji skupień np. wykorzystujących pojęcie centroidu.

wykorzystującej najrzadziej występujące deskryptory w obrębie grupy, często reprezentant odnosił się wyłącznie do jednego lub dwóch atrybutów, nie dostarczając informacji o pozostałych. Przykładowy reprezentant miał postać *(data, Dec 22 2010)(data, Dec 24 2010)(data, Aug 17 2010)(data, Nov 14 2010)(data, Dec 30 2010)(data, Sep 17 2010)(data, Jun 17 2010)* co oznacza, że wśród odkrytej grupy komórek, istnieją takie, które były niedostępne w tych dniach, ale były to odosobnione przypadki. Oceniając samego reprezentanta, nie da się jednak stwierdzić czym jeszcze charakteryzuje się wygenerowane skupienie.

Jednakże nawet relatywnie prosty w interpretacji reprezentant grupy jest niewystarczający, aby na jego podstawie odkryć nową wiedzę (na temat utworzonych skupień). Dlatego też proponuje się rozszerzenie opisu skupień o elementy statystyki opisowej (jak wartości średnie, minimalne, maksymalne itp. interesujących analityka cech). Wówczas, podczas analizy konkretnego skupienia, analityk danych może ocenić centralną tendencję czy rozkład poszczególnych cech, dzięki czemu uzyskuje wgląd w strukturę grupy i potencjalnie łatwiej może ustalić dlaczego został utworzony taki wzorzec (skupienie), bądź porównać go z innymi. Oczywiście to do analityka należy decyzja, na jakich parametrach aktualnie chce skupić swoją uwagę oraz jakie cechy są w danym momencie najbardziej interesujące. Tak stworzony opis skupień będzie integralną częścią wizualizacji struktury grup za pomocą techniki tzw. map prostokątów (ang. *treemaps*). Szczegółowe informacje dotyczące techniki map prostokątów oraz jej wykorzystania w procesie graficznej analizy eksploracyjnej zawiera rozdział 6. Dane odnośnie wykorzystania prezentowanej opisowo–eksploracyjnej reprezentacji skupień w działającym systemie do odkrywania wiedzy z danych złożonych zamieszczone zostały w rozdziale 7.

4.4 Podsumowanie

W niniejszym rozdziale zaprezentowano oraz przeanalizowano statystyki opisu danych (zarówno centralnej tendencji jak i rozproszenia), wykorzystywane w procesie analizy eksploracyjnej. Każda z miar centralnej tendencji dostarcza innego rodzaju wiedzy, przez co powinny one być wyznaczane jednocześnie oraz uznawane jako wzajemnie uzupełniające się, aby prowadzone badania były skuteczne. Dokonano również przeglądu najpopularniejszych metod graficznych, wykorzystujących naturalne predyspozycje kognitywne człowieka oraz uzasadniono ich przydatność w kontekście oceny rozkładu danych, jak również prostej metody wykrywania odchyleń już na początkowym etapie analizy. Ponadto przedstawiono koncepcję opisu skupień, wykorzystującą omówione statystyki opisowe, która znajdzie bezpośrednie zastosowanie w autorskim narzędziu (omówionym w rozdziale 7) do wydobywania wiedzy z danych rzeczywistych.

Rozdział 5

Grupowanie danych oparte na pojęciu gęstości

Wydobywanie wiedzy z rzeczywistych, złożonych baz wiedzy jest procesem wieloetapowym i stawia szereg wymogów wobec algorytmów grupowania jak: możliwość odkrywania skupień o różnej strukturze, odporność na występowanie obiektów izolowanych, posiadanie relatywnie niskiej złożoności obliczeniowej i zajętości pamięci, jasno określone kryteria stopu algorytmu oraz wysoka jakość tworzonych skupień. Niestety klasyczne algorytmy analizy skupień (hierarchiczne i niehierarchiczne) nie spełniają wszystkich wymienionych wymagań.

Wykorzystanie konwencjonalnych algorytmów k-optymalizacyjnych (*niehierarchicznych*) narzuca konieczność określenia przez użytkownika liczby skupień k, na jaką należy podzielić zbiór danych [34]. Implikuje to potrzebę posiadania szerokiej wiedzy dziedzinowej oraz dobrej znajomości struktury danych wejściowych, ponieważ w przeciwnym przypadku prawidłowe ustawienie tego parametru jest praktycznie niemożliwe. Najczęściej wymienianym w literaturze przedmiotu [64] rozwiązaniem tego problemu jest kilkukrotne uruchomienie algorytmu niehierarchicznego, z różnymi wartościami parametru k oraz wybór takiej wartości, dla której został osiągnięty najlepszy jakościowo podział, zgodnie z ustalonym kryterium oceny (np. miarą Total Cost). Jest to jednak zadanie czasochłonne, szczególnie w kontekście grupowania dużych wolumenów danych złożonych. Kolejną wadą algorytmów niehierarchicznych, wynikającą bezpośrednio z ich budowy i sposobu działania, jest losowe generowanie początkowych reprezentantów przyszłych skupień. W dalszych etapach wszystkie obiekty przypisywane są do tego skupienia, do którego reprezentanta są najbardziej podobne. Zatem po każdorazowym uruchomieniu algorytmu, otrzymane wyniki mogą się znacząco różnić. Taki sposób postępowania warunkuje również wrażliwość na występowanie obiektów izolowanych, czy tendencję do odkrywania skupień o sferycznym kształcie. Spotykane w literaturze podejścia do tworzenia optymalnych reprezentantów ograniczają się najczęściej do wielokrotnego zastosowania algorytmu i oceny rezultatów, wykorzystania algorytmu hierarchicznego do ich wyznaczenia, bądź dokonania deterministycznego wyboru zgodnie z przyjętą heurystyką (np. poprzez dobór

skrajnie niepodobnych do siebie reprezentantów). Niestety przy dużych, rzeczywistych zbiorach danych, bezpośrednie zastosowanie pierwszej koncepcji może być niemożliwe (ze względu na wymogi pamięciowe algorytmu), a drugie podejście jest silnie zależne od wybranej heurystyki i może faworyzować obiekty odstające.

Algorytmy hierarchiczne prowadzą do powstania hierarchii skupień z monotonicznie wzrastającym współczynnikiem ich podobieństwa. Ze względu na sposób tworzenia skupień można wyróżnić dwa rodzaje technik hierarchicznych: aglomeracyjne i deglomeracyjne (podziałowe). Podejście aglomeracyjne zakłada, że w pierwszym kroku wszystkie obiekty zbioru danych wejściowych tworzą osobne skupienia. Następnie są one iteracyjnie łączone (biorąc pod uwagę ich podobieństwo) w tzw. grupy wyższego rzędu. Postępowanie zostaje zakończone w momencie otrzymania jednej grupy, agregującej wszystkie obiekty analizowanego zbioru. Techniki podziałowe natomiast traktują cały zbiór jako jedno skupienie i w sposób sekwencyjny dokonują jego dekompozycji, aż do otrzymania skupień jednoelementowych. Powstała struktura może być wizualizowana w postaci tzw. dendrogramu, czyli drzewiastego diagramu ukazującego związki pomiędzy wybranymi elementami (obiektami lub grupami) na podstawie przyjętego kryterium (podobieństwa). Algorytmy hierarchiczne pozbawione są głównej wady podejść niehierarchicznych – nie wymagają określenia a priori docelowej liczby grup. Jednakże ich istotną wadą jest relatywnie wysoka złożoność pamięciowa i zajętość pamięci, ponieważ w klasycznej wersji wykonywanych jest $n-1$ iteracji, a dla każdej z nich tworzona jest macierz (podobieństwa) o wymiarach $n \times n$ (gdzie n to liczba obiektów). Naturalnie najczęściej stosuje się zmodyfikowane podejście, powodujące w każdej iteracji przeliczenie tylko jednej kolumny (dla nowo tworzonej grupy) zamiast liczenia pełnej macierzy. Ponadto wyniki grupowania są zależne od przyjętej techniki łączenia skupień (pojedynczego, średniego lub całkowitego wiązania)– otrzymana struktura może być inna dla każdej techniki. W szczególności metoda pojedynczego wiązania (ang. *single linkage*) wykazuje dużą skłonność do łańcuchowania [43, 34]. Warto również nadmienić, że autor rozprawy dokonał już na etapie wcześniejszych prac [44] analizy algorytmów niehierarchicznych oraz ich porównania z algorytmem AHC (ang. ***A**gglomerative **H**ierarchical **C**lustering*) w zastosowaniu grupowania prac dyplomowych[1]. Ponadto w pracy [48] przedstawiono podejścia stosowane do grupowania dużych wolumenów danych jak próbkowanie danych, dyskretyzacja, metoda dziel i zwyciężaj itp., wraz z przykładami algorytmów, które je wykorzystują oraz zestawieniem wad czy zalet.

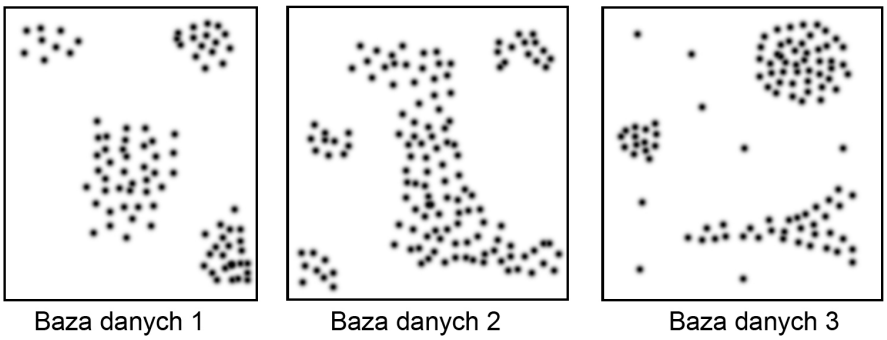
W świetle przedstawionych argumentów, interesujące wydały się gęstościowe techniki analizy skupień. Przedstawicielem tej kategorii jest algorytm DBSCAN (*ang. **D**ensity **B**ased **S**patial **C**lustering of **A**pplications with **N**oise*). Został on opracowany w 1996 roku w Monachium przez czterech badaczy: Martina Estera, Hansa-Petera Kriegela, Jörga Sandera oraz Xiaowei Xu. Pierwotnie był on dedykowany zastosowaniu w tzw. systemach przestrzennych baz danych (ang. *spatial database systems*), służących do zarządzania, przechowywania i wyszukiwania informacji powiązanych z obiektami i zjawiskami w przestrzeni. Jednakże dzięki prostej i skutecznej idei możliwe jest jego bezpośrednie zastosowanie dla danych dowolnego typu. Jego budowa opiera się na idei skupienia traktowanego jako zbiór obiektów o podobnej gęstości, dzięki czemu jest on zdolny odkrywać skupienia o różnej złożoności i kształcie. Gęstość

[1] Szczegóły dotyczące budowy i działania algorytmu AHC znajdują się m.in. w [7].

obiektów odpowiada naturalnie pojmowanemu podobieństwu między nimi. Jest to analogia do tworzenia tzw. map samoorganizujących się [26]. W przytoczonej technice, obiekty reprezentowane są najczęściej jako heksagonalne lub prostokątne węzły grafu, osadzone w przestrzeni dwuwymiarowej[2]. Dwa węzły umieszczone na mapie w bliskiej odległości świadczą o dużym podobieństwie między obiektami. Większa odległość między tymi węzłami oznacza, że obiekty te są do siebie mniej podobne. Metody gęstościowe mają zatem na celu znajdowanie obiektów gęsto ułożonych. Definicja *skupienia* opiera się na obiektach wzajemnie osiągalnych lub połączonych z pewną ustaloną gęstością (co zostanie szczegółowo przedstawione w następnym podrozdziale). DBSCAN umożliwia sterowanie tym zagęszczeniem tworzonych grup obiektów, poprzez dwa parametry wejściowe (promień sąsiedztwa *Eps* oraz minimalną liczbę obiektów grupy *MinPts*).

5.1 Gęstościowa definicja skupienia

Gęstościowa definicja grupy wzorowana jest na naturalnych predyspozycjach człowieka do identyfikowania kształtów i gęsto upakowanych obszarów jako skupienie. Rysunek 5.1 przedstawia trzy różne, przykładowe zbiory danych. Dzięki wrodzonym zdolnościom kognitywnym człowieka, ludzkie oko bez problemu rozpozna wśród każdego ze zbiorów odpowiednie liczby oraz strukturę skupień. W przypadku pierwszej oraz drugiej bazy danych mamy do czynienia z czterema skupieniami, natomiast w ostatnim zbiorze danych występują trzy grupy i szum informacyjny. Głównym powodem, dla którego potrafimy w tak prosty i jednoznaczny sposób zidentyfikować te wzorce jest fakt, że każde skupienie posiada pewne zagęszczenie obiektów, które jest wyraźnie większe, niż zagęszczenie obiektów poza nim. Ponadto stopień gęstości szumu informacyjnego jest wyraźnie niższy, niż gęstość którejkolwiek z grup. Takie gęstościowe pojmowanie skupienia stanowi również analogię do testu gęstości, wykonywanym w algorytmie Rocchia, którego skuteczność potwierdzono praktycznie tworząc system SMART Saltona [55]. Opierając się na przedstawionych zależnościach, dokonano sformalizowania intuicyjnie rozumianych pojęć skupienia czy szumu.



Rysunek 5.1: Trzy przykładowe zbiory danych o różnych gęstościach.
Źródło: Opracowanie własne

[2]Jak zaznaczono w [34], w najnowszych implementacjach techniki SOM odchodzi się od reprezentacji na płaszczyźnie i wykorzystuje się przestrzeń trójwymiarową, by uniknąć zniekształceń w wizualizacji skupień.

Główna idea jest następująca: dla każdego obiektu w danej grupie, sąsiedztwo o podanym promieniu musi zawierać co najmniej minimalną, określoną liczbę obiektów tj. zagęszczenie sąsiedztwa musi przekraczać pewien próg. Kształt sąsiedztwa[3] jest określony przez dobór metryki podobieństwa (często mierzonego za pomocą odległości) dwóch obiektów p i q (oznaczonej jako $dist(p, q)$). Przykładowo, jeśli zostanie użyta miara miejska jako funkcja podobieństwa, to sąsiedztwo będzie miało prostokątny kształt (dla przestrzeni dwuwymiarowej).

Definicja 1 (Sąsiedztwo Eps obiektu)
Sąsiedztwo Eps obiektu p (ozn. $N_{Eps}(p)$) jest określone następująco:

$$N_{Eps}(p) = q \in D | dist(p, q) \leqslant Eps,$$

gdzie D to zbiór danych, $dist(p, q)$ podobieństwo między obiektami p i q, natomiast Eps to maksymalny promień sąsiedztwa. Sąsiedztwo Eps obiektu p stanowią zatem wszystkie te obiekty q, których odległość od obiektu p jest mniejsza niż założona wartość promienia Eps.

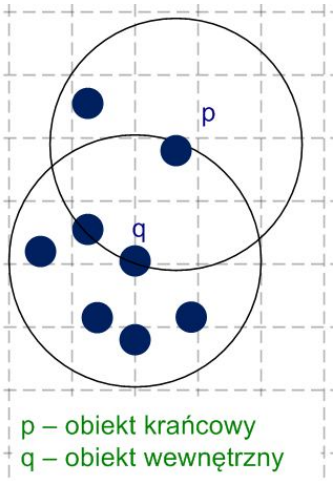
Sama definicja sąsiedztwa obiektu nie jest jednak wystarczająca do określenia pojęcia skupienia, ponieważ może to prowadzić do błędnego wniosku, że dla każdego obiektu w grupie, w jego *Eps-sąsiedztwie* musi znajdować się co najmniej minimalna liczba (*MinPts*) obiektów. Takie rozumowanie jest błędne, gdyż wyróżnia się dwa rodzaje obiektów w grupie: obiekty wewnątrz skupienia, blisko jego jądra (*tzw. core points*) oraz obiekty znajdujące się na granicy skupienia (*tzw. border points*). Sąsiedztwo takiego krańcowego obiektu zawiera wyraźnie mniej obiektów, aniżeli sąsiedztwo obiektu wewnętrznego. Należałoby zatem ustawić wartość parametru minimalnej liczby obiektów na relatywnie małą, by uwzględnić wszystkie obiekty wchodzące w skład danego skupienia. Niestety taka stała wartość nie byłaby właściwa w odniesieniu do każdej grupy, zwłaszcza w przypadku występowania wartości izolowanych. Wymaga się zatem, by dla każdego obiektu p w skupieniu C, istniał taki obiekt q należący do C, żeby p znajdował się w sąsiedztwie Eps q oraz $N_{Eps}(q)$ zawierało co najmniej $MinPts$ obiektów (co zostało wyrażone przez definicję nr 2).

Definicja 2 (Bezpośrednia gęstościowa osiągalność)
Obiekt p jest bezpośrednio gęstościowo osiągalny z obiektu q jeżeli:

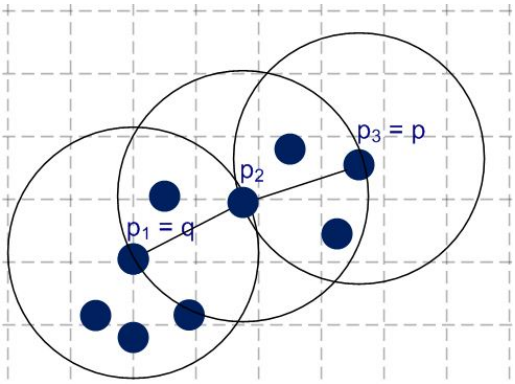
1. $p \in N_{Eps}(q)$ oraz
2. $|N_{Eps}(q)| \geqslant MinPts$ (warunek na obiekt wewnętrzny).

Naturalnie, bezpośrednia gęstościowa osiągalność jest symetryczna dla par obiektów wewnętrznych oraz niesymetryczna dla pary obiektów zewnętrznego i wewnętrznego. Rysunek 5.2 ilustruje ten drugi przypadek (przy założeniu, że $MinPts = 4$). Z przytoczoną definicją związana jest kolejna, bardziej ogólna definicja gęstościowej osiągalności.

Definicja 3 (Gęstościowa osiągalność)
Obiekt p jest gęstościowo osiągalny z obiektu q, jeżeli istnieje taki łańcuch $p_1, \ldots, p_n, p_1 = q, p_n = p$ obiektów, że p_{i+1} jest gęstościowo bezpośrednio osiągalny z p_i.



Rysunek 5.2: Bezpośrednia gęstościowa osiągalność jako relacja niesymetryczna.
Źródło: Opracowanie własne

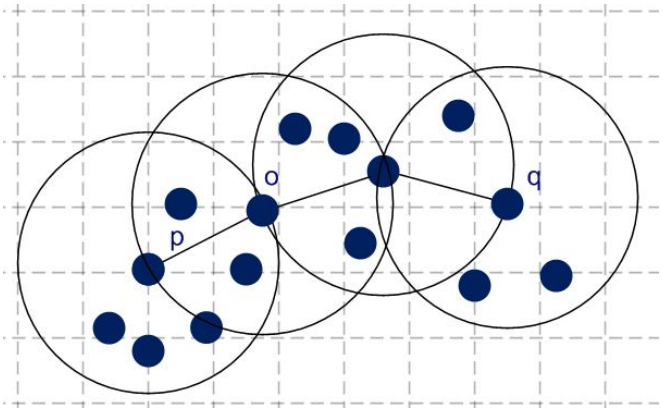


Rysunek 5.3: Gęstościowa osiągalność.
Źródło: Opracowanie własne

Relacja przedstawiona w definicji 3 jest przechodnia, ale nie jest symetryczna. Rysunek 5.3 przedstawia przypadek asymetryczny dla gęstościowej osiągalności (przy $MinPts = 4$). Mimo, że ogólnie rzecz biorąc relacja ta nie jest symetryczna, to jednak w przypadku obiektów wewnętrznych zachowuje ona symetryczność. Dwa obiekty krańcowe tego samego skupienia C mogą nie być gęstościowo osiągalne od siebie, ponieważ warunek $|N_{Eps}(q)| \geqslant MinPts$ może nie zostać spełniony. Jednakże, musi istnieć obiekt wewnętrzny w skupieniu, z którego oba obiekty krańcowe byłyby gęstościowo osiągalne. Dlatego też stworzono kolejną definicję połączenia gęstościowego. Połączenie gęstościowe (przedstawione na rysunku 5.4) jest relacją symetryczną, spełnioną nawet dla obiektów krańcowych danego skupienia.

Definicja 4 (Połączenie gęstościowe)
Obiekt p jest połączony gęstościowo z obiektem q, jeżeli istnieje taki obiekt o, że oba obiekty p i q są gęstościowo osiągalne z o.

[3]Pojęcie sąsiedztwa jest zgodne z definicją nr 1.



Rysunek 5.4: Połączenie gęstościowe.
Źródło: Opracowanie własne

Biorąc pod uwagę przytoczone definicje można przystąpić do zdefiniowania pojęcia skupienia, opartego na idei gęstości. Intuicyjnie grupa jest maksymalnym zbiorem gęstościowo połączonych obiektów. Natomiast szum informacyjny jest zbiorem obiektów nienależących do żadnego skupienia. Formalnie pojęcia te przedstawiono za pomocą definicji nr 5 oraz 6. Należy zauważyć, że skupienie C zawiera co najmniej $MinPts$ obiektów.

Definicja 5 (Skupienie)
Niech D będzie zbiorem obiektów, który chcemy podzielić na grupy. Skupieniem nazywamy niepusty podzbiór D spełniający warunki:

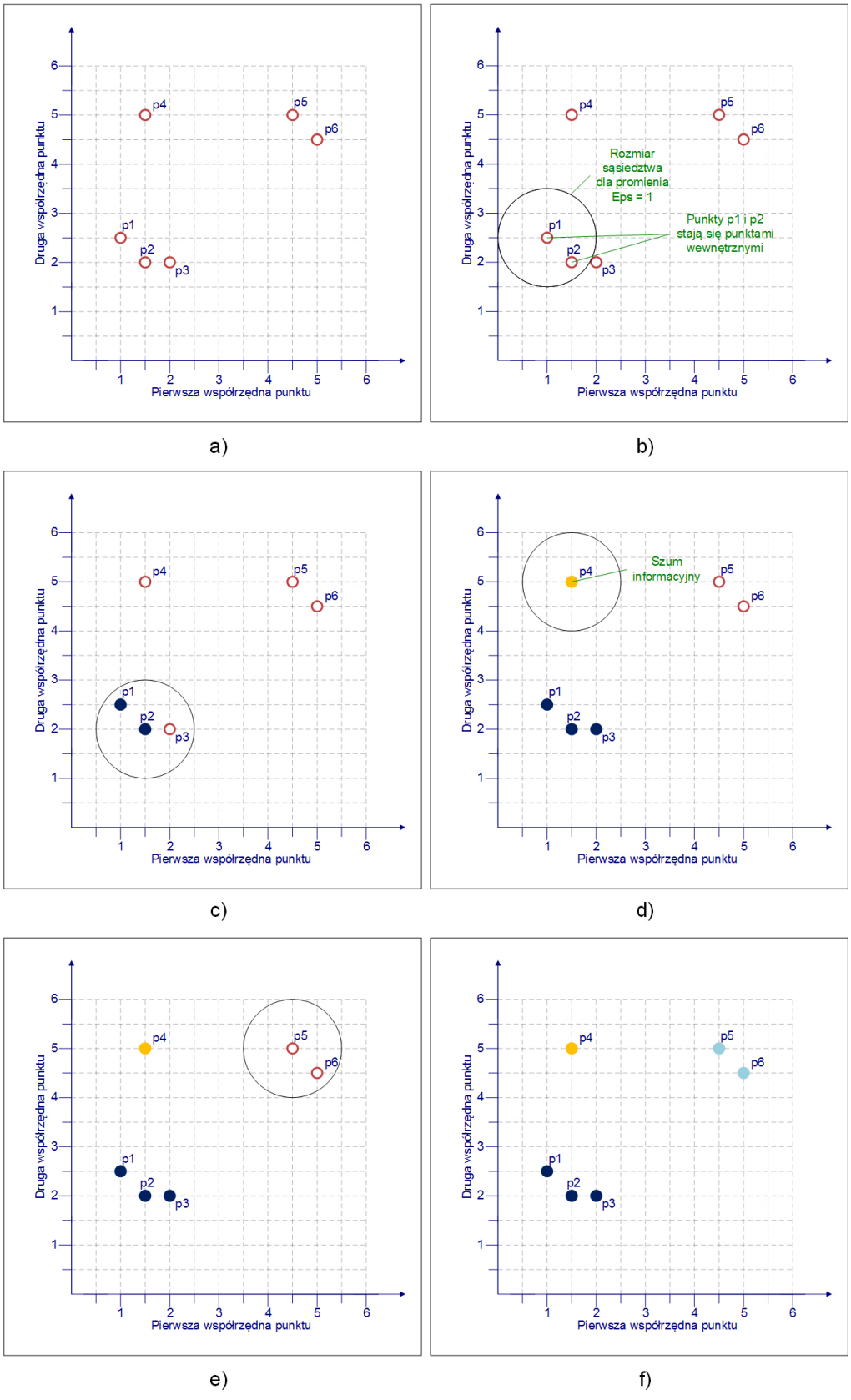
1. $\forall p, q$: jeżeli $p \in C$ oraz q jest gęstościowo osiągalne z p to $q \in C$.
2. $\forall p, q \in C$: p jest gęstościowo połączony z q.

Definicja 6 (Szum informacyjny)
Niech $C_1, \ldots, C_k$ będą skupieniami w zbiorze obiektów D przy określonych parametrach Eps_i oraz $MinPts_i$, $i = 1, \ldots, k$. Szum informacyjny jest zatem podzbiorem obiektów z D nie należącym do żadnego skupienia C_i, to jest:

$$szum = p \in D | \forall i : p \notin C_i.$$

5.2 Analiza algorytmu DBSCAN

Do poprawnego działania algorytm gęstościowy DBSCAN wymaga określenia wartości dwóch parametrów: maksymalnego promienia sąsiedztwa (Eps) oraz minimalnej liczby obiektów wchodzących w skład odkrywanych skupień ($MinPts$). W najlepszym możliwym przypadku parametry te byłyby dobierane adaptacyjnie, dla każdego skupienia z osobna. Wówczas uzyskany podział można nazwać optymalnym (dla wykorzystywanego podejścia). Nie jest to jednak



Rysunek 5.5: Przykład działania algorytmu DBSCAN.
Źródło: Opracowanie własne

rzecz trywialna, dlatego algorytm wykorzystuje globalne wartości tych parametrów, wspólne dla wszystkich obiektów i skupień. W tym celu została zdefiniowana przez autorów algorytmu prosta heurystyka (omówiona w dalszej części niniejszej pracy), która pozwala określić wartości szukanych parametrów dla najmniej zagęszczonego skupienia. Ustala ona najmniejsze zagęszczenie, dla którego obiekty nie są jeszcze klasyfikowane jako szum informacyjny.

Pierwszym krokiem algorytmu jest wylosowanie obiektu p oraz wyznaczenie wszystkich obiektów, które są gęstościowo osiągalne z p (przy określonych wartościach *Eps* i *MinPts*). Jeżeli p jest obiektem wewnętrznym, to krok ten skutkuje powstaniem pierwszej grupy. Jeżeli p jest obiektem krańcowym, to żaden obiekt nie jest gęstościowo osiągalny z p, więc algorytm wybiera kolejny obiekt ze zbioru danych. Proces ten jest powtarzany, aż nie zostaną przeanalizowane wszystkie obiekty ze zbioru danych wejściowych. Obiekty niezakwalifikowane do żadnego skupienia są oznaczane jako szum informacyjny.

Listing 5.1: Algorytm DBSCAN

```
// ZbiorObiektow ma status NIESKLASYFIKOWANY
DBSCAN (ZbiorObiektow, Eps, MinPts)
SkId := nextId(SZUM);
FOR i FROM 1 TO ZbiorObiektow.size DO
  Obiekt := ZbiorObiektow.get(i);
  IF Obiekt.SkId = NIESKLASYFIKOWANY THEN
   IF RozszerzSkupienie(ZbiorObiektow, Obiekt, SkId, Eps, MinPts) THEN
     SkId := nextId(SkId)
   END IF
  END IF
END FOR
END; // DBSCAN
```

Przedstawiona idea grupowania gęstościowego zostanie zilustrowana na przykładzie. Rysunek 5.5a prezentuje sztucznie wygenerowany zbiór danych dwuwymiarowych, składający się z sześciu obiektów (reprezentowanych na rysunkach jako zestaw punktów). Niech zastosowaną miarą podobieństwa obiektów będzie miara euklidesowa i zostaną ustalone następujące wartości parametrów wejściowych algorytmu: $Eps = 1$, $MinPts = 2$. Na początku algorytm wybiera losowo obiekt ze zbioru danych (np. *p1*). Następnie wyznaczane jest sąsiedztwo wybranego obiektu zgodnie z określoną miarą podobieństwa i wartościami parametrów startowych. Ilustruje to rysunek 5.5b. Ponieważ w sąsiedztwie obiektu *p1* znajduje się obiekt *p2* oraz jest spełniony warunek na obiekt wewnętrzny (zgodnie z definicją nr 2) tworzone jest pierwsze skupienie zawierające wspomniane obiekty (oznaczone kolorem granatowym na rysunku 5.5c). Algorytm następnie rozpoczyna analizę obiektu *p2* poprzez wyznaczenie jego sąsiedztwa. W sąsiedztwie aktualnie analizowanego obiektu znajduje się obiekt *p3*, który również spełnia warunek na obiekt wewnętrzny, więc jest on dodawany do wcześniej utworzonego skupienia. W dalszej części DBSCAN analizuje kolejny, dotychczas nieprzetworzony obiekt czyli *p4*. Zgodnie z rysunkiem 5.5d jego sąsiedztwo nie zawiera żadnego innego obiektu, wobec czego niespełniony jest warunek na minimalną liczbę obiektów wchodzących w skład danej grupy i uznawany jest on za szum informacyjny. Algorytm ponownie wyznacza kolejny, nieprzeanalizowany obiekt ze zbioru danych czyli *p5*. Obiekt *p5* jest obiektem wewnętrznym, więc tworzona jest druga

grupa zawierająca obiekty *p5* oraz *p6*. Jako, że wszystkie punkty ze zbioru danych zostały już zbadane, algorytm kończy swoje działanie. Podstawowa wersja algorytmu DBSCAN zapisana w pseudokodzie została dodatkowo zaprezentowana na listingu 5.1.
ZbiorObiektow na listingu 5.1 jest utożsamiany z całym zbiorem zawartym w bazie danych. *Eps* i *MinPts* są globalnymi parametrami ustalonymi przez użytkownika lub na podstawie stosownej heurystyki, sterującymi spójnością i liczbą obiektów wewnątrz skupień. Funkcja *ZbiorObiektow.get(i)* zwraca *i*-ty element należący do struktury *ZbiorObiektow*. Najważniejszą funkcją algorytmu jest funkcja *RozszerzSkupienie*, której rozwinięcie zostanie zaprezentowane poniżej (na listingu 5.2).

Listing 5.2: Funkcja RozszerzSkupienie

```
RozszerzSkupienie(ZbiorObiektow, Obiekt, SkId, Eps, MinPts): Boolean;
seeds:=ZbiorObiektow.regionQuery(Obiekt,Eps);
IF seeds.size<MinPts THEN // Obiekt nie jest obiektem wewnętrznym
  ZbiorObiektow.changeSkId(Obiekt,SZUM);
  RETURN False;
ELSE // wszystkie obiekty w seeds są gęstościowo osiągalne z Obiekt
  ZbiorObiektow.changeSkId(seeds,SkId);
  seeds.delete(Obiekt);
  WHILE seeds <> Empty DO
    currentP := seeds.first();
    result := ZbiorObiektow.regionQuery(currentP, Eps);
    IF result.size >= MinPts THEN
      FOR i FROM 1 TO result.size DO
        resultP := result.get(i);
        IF resultP.SkId IN {NIESKLASYFIKOWANY, SZUM} THEN
          IF resultP.SkId = NIESKLASYFIKOWANY THEN
            seeds.append(resultP);
          END IF;
          ZbiorObiektow.changeSkId(resultP,SkId);
        END IF;
      END FOR;
    END IF;
    seeds.delete(currentP);
  END WHILE;
  RETURN True;
END IF
END; // RozszerzSkupienie
```

Pierwszym krokiem realizowanym w zaprezentowanej funkcji jest wyznaczenie sąsiedztwa Eps dla aktualnie analizowanego obiektu. Odpowiada za to funkcja *SetOfPoints.regionQuery(Point,Eps)*, która zwraca listę (oznaczoną jako *seeds*) obiektów będących w sąsiedztwie, o promieniu *Eps*.

Lista *seeds* zawiera kandydatów do włączenia w ewentualne skupienie, dlatego w dalszym postępowaniu następuje sprawdzenie, czy liczy ona co najmniej *MinPts* obiektów. Jeśli nie (czyli występuje zbyt niskie zagęszczenie obiektów), analizowany obiekt jest oznaczany jako szum (poprzez nadanie odpowiedniej wartości polu *SkId*).Właściwość *SkId* (identyfikator skupienia) dla obiektów, które zostały początkowo oznaczone jako szum informacyjny, może zostać

w późniejszym etapie zmieniona, jeśli okaże się, że są one gęstościowo osiągalne z jakiegoś innego obiektu ze zbioru danych. Taka sytuacja może mieć miejsce, jeżeli obiekt początkowo oznaczony jako szum jest w rzeczywistości obiektem (krańcowym) znajdującym się na granicy skupienia.

Jeżeli natomiast poprzedni warunek jest spełniony, to wszystkie obiekty w liście *seeds* są gęstościowo osiągalne z aktualnie analizowanego obiektu. *Obiekt* musi być wewnętrznym i wraz ze swoimi sąsiadami tworzy pierwsze skupienie – każdemu z tych obiektów jest nadawany ten sam numer *SkId* – po czym zostaje on usunięty z listy. Następnie każdy obiekt z *seeds* jest analizowany pod kątem swojego sąsiedztwa Eps, co może skutkować powiększeniem odkrytego skupienia. Jeżeli obiekt, mający zostać wcielony do aktualnego skupienia, miał status *niesklasyfikowany*, to jest on dodawany do listy *seeds* i omówiona procedura jest powtarzana.

Złożoność obliczeniowa i zajętość pamięci

Największy wpływ na czas działania algorytmu DBSCAN, a w konsekwencji na jego złożoność obliczeniową, ma wyliczanie sąsiedztwa Eps, realizowane przez metodę *regionQuery* uruchamianą dla różnych obiektów. W celu optymalizacji tego procesu można wykorzystać strukturę indeksującą R*-drzew[4] (*ang. R*-trees*), do której to wczytywany jest zbiór wejściowy. R*-drzewa, do opisania obiektów wielowymiarowych, wykorzystują minimalne regiony pokrywające (najczęściej w formie prostokątów)[5]. Średnia złożoność obliczeniowa (przy zastosowaniu R*-drzew) pojedynczego wywołania funkcji *regionQuery* to $O(log\ n)$ – gdzie n to liczba obiektów w zbiorze danych – ponieważ wystarczy przejrzeć tylko niewielką liczbę węzłów w R*-drzewie. W najgorszym przypadku, dla każdego obiektu danych, wywołana będzie jednokrotnie wspomniana funkcja. Wynika z tego, że średnia złożoność obliczeniowa dla algorytmu DBSCAN wynosi $O(n \cdot \log n)$ [16]. Warto zauważyć, że w przypadku dużej liczby wymiarów i niewykorzystywania struktur typu R*-drzew, złożoność obliczeniowa algorytmu wzrasta do $O(n^2)$.

Zajętość pamięci algorytmu jest relatywnie mała i wynosi w przybliżeniu $O(n)$, nawet dla danych wielowymiarowych, ponieważ wymagane jest przechowywanie tylko niewielkiej liczby danych dla każdego obiektu, tj. identyfikatora skupienia oraz klasy obiektu (wewnętrzny, krańcowy, szum) [64]. Sytuacja ulega pogorszeniu, jeżeli w celu optymalizacji czasu działania, przechowywana jest w pamięci cała macierz odległości (podobieństwa) – wówczas szacowana zajętość pamięci wynosi $O(n^2)$. Nie jest to jednak zabieg konieczny.

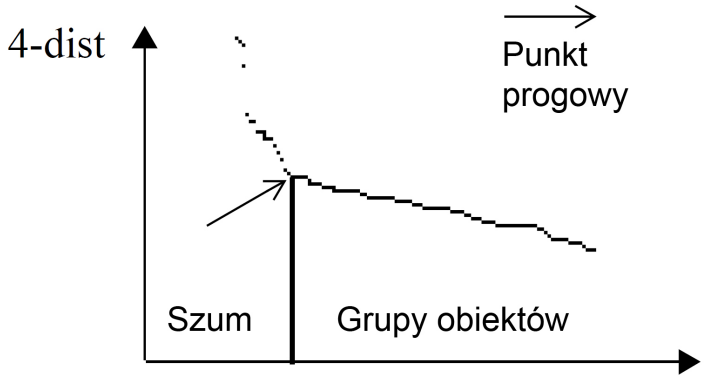
Heurystyka wyznaczania parametrów startowych

Jak zostało wspomniane wcześniej, podczas wyznaczania wartości parametrów *Eps* i *MinPts* można posłużyć się prostą heurystyką, która bierze pod uwagę najcieńsze skupienie w zbiorze. Jej idea jest wynikiem następującej obserwacji. Niech b będzie odległością (w sensie podobieństwa) między obiektem p a jego k-tym najbliższym sąsiadem. Wówczas b-sąsiedztwo dla obiektu p zawiera dokładnie $k + 1$ obiektów. Tylko w przypadku, gdy kilkanaście obiektów

[4]Symbol gwiazdki w zapisie utożsamiany jest z liczbą wymiarów zbioru wejściowego. Przykładowo R^2-drzewa określają strukturę do przechowywania danych dwuwymiarowych.

[5]Więcej informacji na temat działania i wykorzystania R*-drzew znajduje się w [42].

ma dokładnie taką samą odległość od p, b-sąsiedztwo zawiera więcej niż $k+1$ elementów. Ponadto załóżmy, że dla danego k została zdefiniowana funkcja *k-dist*, która odwzorowuje każdy obiekt w bazie danych na odległość do jego k-tego najbliższego sąsiada. Po posortowaniu obiektów w porządku malejącym na podstawie wartości *k-dist*, jej wykres (*ang. sorted k-dist graph*) pozwoli uzyskać informacje o przybliżonym rozkładzie zagęszczenia danych. Po uprzednim losowym doborze obiektu p oraz ustaleniu wartości parametru *Eps* na *k-dist(p)*, natomiast wartości *MinPts* na k, wszystkie obiekty z przyporządkowaną mniejszą (lub równą) wartością *k-dist* staną się obiektami wewnętrznymi. W celu wyznaczenia optymalnej wartości *Eps*, należy zatem określić punkt progowy (*ang. threshold point*) z maksymalną wartością *k-dist* dla najmniej zagęszczonego skupienia. Jest to pierwszy punkt, w pierwszej dolinie znalezionej na wykresie *k-dist* (jak zaprezentowano na rysunku 5.6). Wszystkie obiekty położone na lewo do punktu progowego są uznawane za szum, natomiast wszystkie pozostałe obiekty są przypisywane do jakieś grupy.

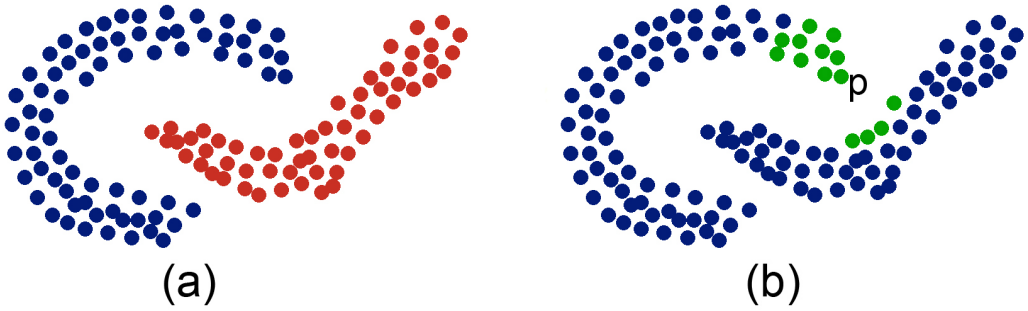


Rysunek 5.6: Wyznaczanie punktu progowego.
Źródło: [16]

Pozostaje jednak problem wyznaczenia liczby sąsiadów k. Autorzy algorytmu DBSCAN w [16] sugerują ustalenie parametru k na 4 dla wszystkich dwuwymiarowych baz danych, ponieważ z przeprowadzonych testów wynika, że wykresy *k-dist* dla większych wartości nie różnią się znacząco. W kontekście analizy wielowymiarowych danych złożonych, takie rozwiązanie jest nieakceptowalne i dobór wartości parametrów startowych algorytmu DBSCAN, powinien uwzględniać posiadaną wiedzę dziedzinową na temat analizowanego zbioru, jak również inne kryteria umożliwiające ocenę jakości danego podziału na grupy.

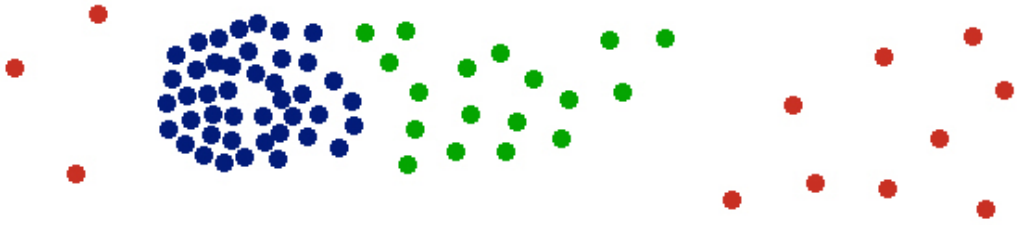
Dobór parametrów startowych (a w szczególności promienia sąsiedztwa *Eps*) ma znaczący wpływ na jakość ostatecznego podziału na skupienia. Jeżeli naturalnie występujące grupy charakteryzują się zmienną gęstością, algorytm DBSCAN może zaklasyfikować je w sposób nieprawidłowy. Przykładowo dwie mniejsze grupy o różnych zagęszczeniach obiektów, mogą zostać uznane za jedno większe skupienie. Sytuację tę ilustruje rysunek 5.7. Algorytm błędnie sklasyfikował obiekty jako jedno skupienie (rys. 5.7 b), podczas gdy powinny one zostać rozdzielone na dwie grupy (rys. 5.7 a). Dzieje się tak, ponieważ algorytm analizując obiekt p, mylnie uznaje go za wewnętrzny, powiększając jego sąsiedztwo (które zaznaczono na zielono).

W konsekwencji obiekt p stanowi pomost między dwoma skupieniami, a algorytm dokonuje błędnego przyporządkowania.



Rysunek 5.7: Problem właściwej identyfikacji skupień o różnych zagęszczeniach.
Źródło: Opracowanie własne

Równie niekorzystna sytuacja występuje, jeżeli wartość parametru *Eps* będzie zbyt niska. Wówczas obiekty, które powinny zostać uznane za część skupienia, klasyfikowane są jako szum informacyjny. Przedstawiono to na rysunku 5.8. Obiekty zaznaczone na zielono zostały zaklasyfikowane jako szum informacyjny, podczas gdy powinny one należeć do niebieskiego skupienia. Taki stan rzeczy ma miejsce, ponieważ skupienie zielone charakteryzuje się zmienną gęstością należących do niego elementów. Elementy prawidłowo zaklasyfikowane jako szum informacyjny, oznaczono kolorem czerwonym.



Rysunek 5.8: Błędna identyfikacja skupień przy zbyt niskiej wartości Eps.
Źródło: Opracowanie własne

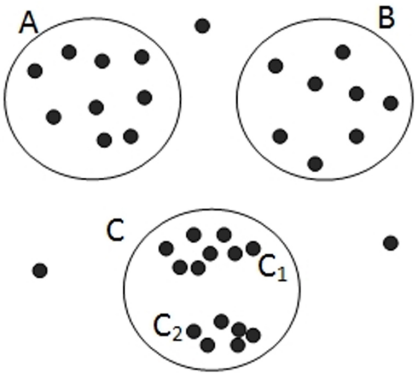
Zalety i wady algorytmu DBSCAN

Bezsprzeczną zaletą algorytmu DBSCAN (jak i wszystkich opartych na pojęciu gęstości) jest możliwość odkrywania skupień o różnych (często nieregularnych) kształtach. Kolejną równie cenną zaletą tego algorytmu jest jego odporność na występowanie wartości izolowanych (szumu informacyjnego). Ponadto DBSCAN, w przeciwieństwie do algorytmów k-optymalizacyjnych, nie wymaga wcześniejszego określenia liczby skupień, na jakie zostanie podzielony zbiór danych. Nie bez znaczenia jest również niska złożoność obliczeniowa i zajętość pamięci (przy zastosowaniu odpowiednich struktur do przechowywania danych).

Do wad opisywanego algorytmu należą: trudność w prawidłowej klasyfikacji zbiorów obiektów o różnych gęstościach oraz konieczność określenia parametrów startowych *MinPts* i *Eps*. Wymaga to bowiem posiadania stosownej wiedzy dziedzinowej lub korzystania z odpowiednich heurystyk. Ponadto zastosowanie konkretnej heurystyki może negatywnie wpłynąć na złożoność obliczeniową algorytmu (dla zaproponowanego wcześniej rozwiązania, zależy to od złożoności algorytmu wyliczania k-najbliższych sąsiadów).

5.3 OPTICS jako gęstościowa metoda analizy struktury danych

Kolejnym algorytmem opartym na idei gęstości, który zostanie omówiony w niniejszej pracy jest algorytm **OPTICS** (*ang.* ***O****rdering* ***P****oints* ***T****o* ***I****dentify the* ***C****lustering* ***S****tructure*). W przeciwieństwie do swojego pierwowzoru (algorytmu DBSCAN), nie generuje on bezpośrednio podziału obiektów na grupy, a jedynie ściśle określone uporządkowanie obiektów, które reprezentuje zagęszczenie elementów ze zbioru danych. Pozwala to ocenić wewnętrzną strukturę danych, jak również dokonać jej wizualizacji. Stanowi on także bezpośrednią odpowiedź na największe wady poprzedniego algorytmu, czyli problematyczny dobór właściwych parametrów startowych oraz możliwość błędnej identyfikacji naturalnych skupień w sytuacji, gdy posiadają one różne zagęszczenia. Został on zaprezentowany przez Mihaela Ankersta, Markusa M. Breuniga, Hansa-Petera Kriegela, Jörga Sandera w 1999 roku.



Rysunek 5.9: Występowanie hierarchii skupień.
Źródło: Opracowanie własne

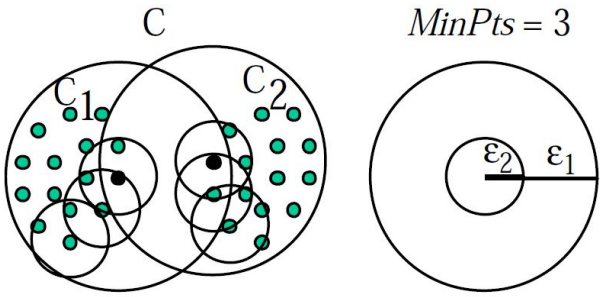
Wewnętrzna struktura danych, a w konsekwencji również utworzonych grup bardzo często nie może zostać właściwie scharakteryzowana przy użyciu jedynie globalnych parametrów dotyczących jej gęstości i rozkładu. Wymagany może być adaptacyjny dobór stopnia zagęszczenia, by odkryć inne rodzaje skupisk (szczególnie analizując dane rzeczywiste). Rysunek 5.9 przedstawia opisywaną sytuację, w której niemożliwe byłoby znalezienie skupień A, B, C_1, C_2 jednocześnie, korzystając wyłączne z jednego globalnego parametru promienia sąsiedztwa. W takim przypadku zostałyby odkryte jedynie grupy A, B, C lub C_1, C_2 (przy czym naturalnie występujące skupienia A i B zostałyby potraktowane jak szum informacyjny). W literaturze przedmiotu [4] spotyka się dwa rozwiązania tego problemu: użycie algorytmów hierarchicznych

(tam gdzie to możliwe) lub algorytmów opartych na idei gęstości, ale z uwzględnieniem różnych wartości parametrów sterujących.

Oba proponowane sposoby mają jednak swoje wady. Algorytmy hierarchiczne wykorzystujące metodę pojedynczego wiązania, nie potrafią dobrze odseparować specyficznych skupień (*tzw. single-link effect*) oraz trudno jest ocenić rezultaty grupowania (przedstawione w postaci dendrogramu) w przypadku sporej liczby (kilkuset) obiektów. Natomiast druga alternatywa wymaga testowania większego zestawu wartości parametrów oraz przechowywania wszystkich pośrednich rezultatów grupowań (bądź dokonywania oceny ich jakości na bieżąco). Mimo wszystko, taka procedura nadal nie gwarantuje, że zostaną odkryte wszystkie występujące skupienia. Dlatego też rozsądnym wydaje się wykorzystanie idei uporządkowanego przeglądu obiektów ze zbioru danych wejściowych, która jest podstawą działania algorytmu OPTICS, gdyż umożliwi to adaptacyjne strojenie parametru promienia sąsiedztwa.

Idea uporządkowania obiektów

Motywacją do stworzenia algorytmu OPTICS, była obserwacja związana ze stopniem zagęszczenia obiektów w sytuacji, gdy występuje hierarchia (zawieranie się) skupień. W zależności od doboru globalnego parametru promienia sąsiedztwa dla algorytmu gęstościowego, może dojść do złączenia dwóch mniejszych, naturalnie występujących skupień. Ilustruje to rysunek 5.10.



Rysunek 5.10: Hierarchia skupień.
Źródło: [4]

Jeżeli wartość *Eps* ustalimy na $\epsilon_2 << \epsilon_1$, zostaną odkryte grupy C_1 i C_2. Jeżeli natomiast wartość parametru *Eps* będzie odpowiednio większa od ϵ_1, to jedyną wygenerowaną grupą będzie grupa C, zawierająca w sobie podgrupy C_1 i C_2. Wykorzystując ten fakt, można by było zmodyfikować algorytm DBSCAN, by zawsze wybierany był najpierw obiekt, który jest gęstościowo-osiągalny przy najmniejszej wartości promienia sąsiedztwa Eps. Wówczas skupienia charakteryzujące się największym zagęszczeniem (mała wartość *Eps*), byłyby wykrywane jako pierwsze.

Algorytm OPTICS działa podobnie do opisanej modyfikacji z tą różnicą, że nie są przypisywane przynależności obiektów do konkretnych grup. Dla każdego obiektu wyznaczana jest tzw. jego wewnętrzna (*ang. core-distance*) oraz osiągalna odległość (*ang. reachability-distance*). Na podstawie tych parametrów, generowany i zapisywany jest porządek, w jakim obiekty są przetwarzane. Mimo, że opisywana technika nie generuje wprost podziału na grupy, wykorzystując

uzyskane informacje, można w relatywnie prosty sposób podzielić elementy zbioru danych na skupienia, co zostanie pokazane w dalszej części niniejszego rozdziału.

Definicja 7 (Odległość wewnętrzna dla obiektu p)
Niech p będzie obiektem ze zbioru danych D, Eps będzie promieniem sąsiedztwa, $N_{Eps}(p)$ będzie sąsiedztwem Eps obiektu p, $MinPts$ liczbą naturalną, a $MinPts - dist(p)$ odległością z p do jego MinPts-najbliższego sąsiada. Wtedy odległość wewnętrzna zdefiniowana jest jako:

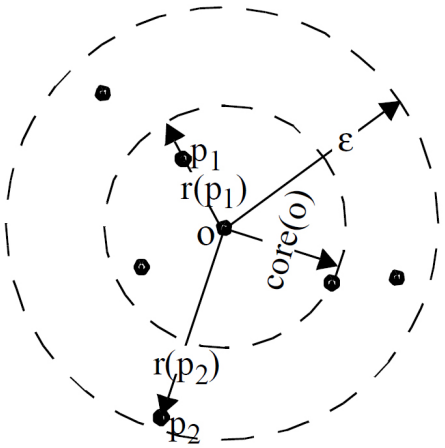
$$\texttt{odlWew}_{Eps,MinPts}(p) = \begin{cases} NIEOKREŚLONA, \textbf{ jeżeli } |N_{Eps}(p)| < MinPts \\ MinPts\text{-}dist(p), \textbf{ w przeciwnym wypadku} \end{cases}$$

Odległość wewnętrzna dla obiektu p, może być utożsamiana z najmniejszą odległością pomiędzy p i obiektem w jego sąsiedztwie, w obrębie którego p byłby klasyfikowany jako obiekt wewnętrzny. W przeciwnym przypadku odległość wewnętrzna jest nieokreślona.

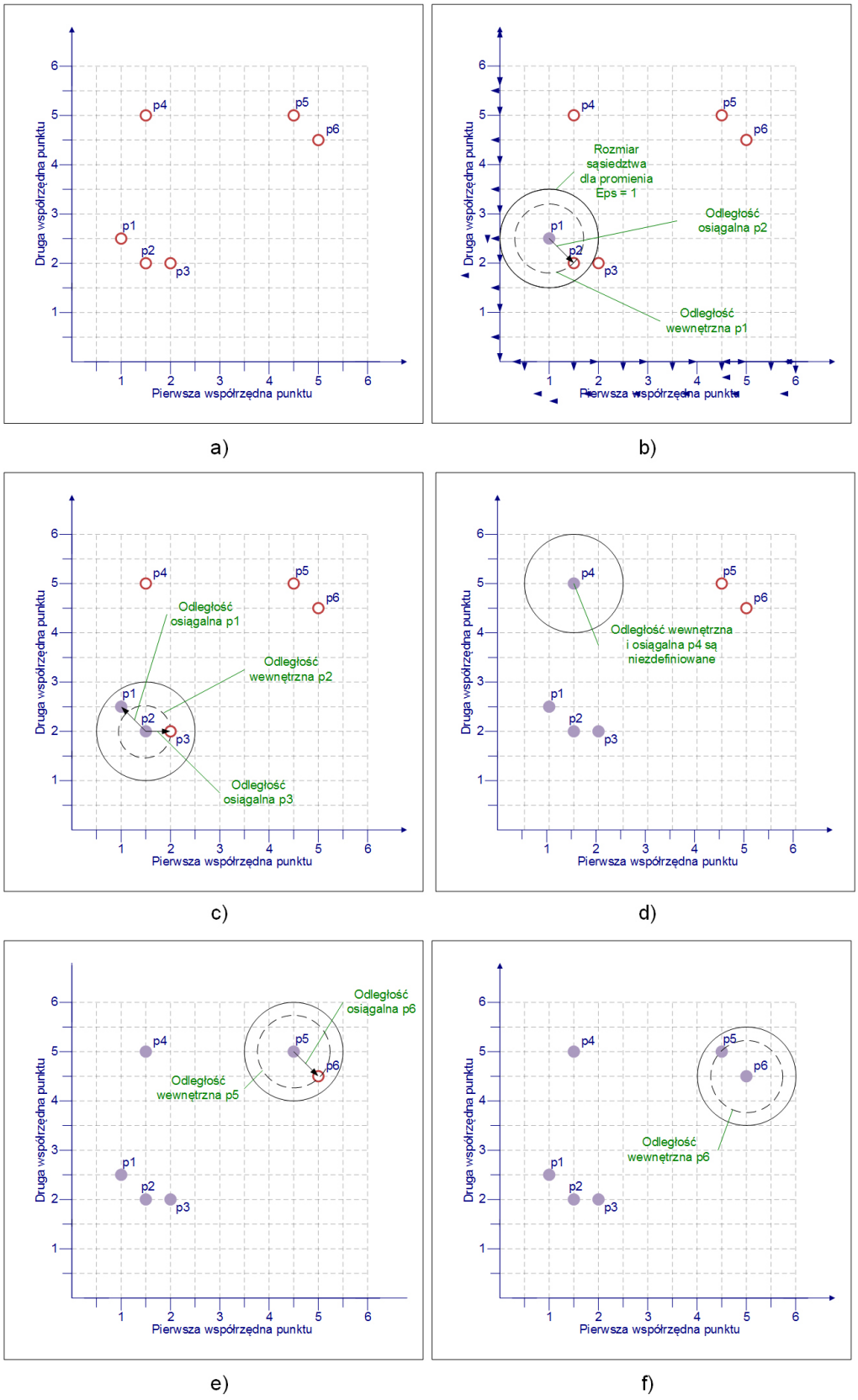
Definicja 8 (Odległość osiągalna dla obiektu p)
Niech p i q będą obiektami ze zbioru danych D, $N_{Eps}(q)$ oznacza sąsiedztwo Eps obiektu q, a $MinPts$ będzie liczbą naturalną. Wówczas odległość osiągalna dla p, w odniesieniu do q, jest zdefiniowana następująco:

$$\texttt{odlOsg}_{Eps,MinPts}(p,q) = \begin{cases} NIEOKREŚLONA, \textbf{ jeżeli } |N_{Eps}(q)| < MinPts \\ \max\{\texttt{odlWew(q)}, \texttt{odległość(q, p)}\}, \textbf{w przeciwnym wypadku} \end{cases}$$

Intuicyjnie rzecz ujmując, odległość osiągalna dla obiektu p w stosunku do obiektu wewnętrznego q, to najmniejsza odległość taka, że p jest bezpośrednio gęstościowo osiągalny z q. W takim przypadku, odległość osiągalna nie może być mniejsza niż odległość wewnętrzna dla q, ponieważ dla mniejszych wartości, p nie jest bezpośrednio gęstościowo osiągalny z q. Jeżeli natomiast q nie spełnia warunku na obiekt wewnętrzny, odległość osiągalna dla obiektu p (w odniesieniu do q) jest nieokreślona. Rysunek 5.11 ilustruje zależności między przytoczonymi definicjami.



Rysunek 5.11: Odległość wewnętrzna *core(o)* oraz odległości osiągalne *r(p1,o)*, *r(p2,o)*.
Źródło: [4]



Rysunek 5.12: Przykład działania algorytmu OPTICS.
Źródło: Opracowanie własne

Zasada działania algorytmu OPTICS

Zasada działania algorytmu OPTICS przedstawiona jest na rysunku 5.12. Podobnie jak dla poprzedniego algorytmu, zbiór danych liczy sześć obiektów reprezentowanych jako punkty (rysunek 5.12a), a parametry wejściowe zostały ustawione na $Eps = 1$, $MinPts = 2$. Pierwszym krokiem algorytmu jest wybór obiektu ze zbioru danych, co może odbywać się losowo bądź deterministycznie (zgodnie z ich kolejnością w zbiorze danych). Niech wybranym obiektem będzie *p1*. Dla *p1* wyznaczane jest jego Eps-sąsiedztwo, odległość wewnętrzna oraz jest on zapamiętany jako przeanalizowany (co zaznaczono fioletowym kolorem na rysunku 5.12b). Jeżeli aktualnie analizowany obiekt jest wewnętrznym, wyliczane są odległości osiągalne dla obiektów w jego sąsiedztwie (względem analizowanego). Następnie uaktualniana jest kolejność analizy obiektów – ten z najmniejszą odległością osiągalną, będzie przetwarzany jako następny. W zaprezentowanej na rysunku 5.12b sytuacji, tylko $p2$ należy do wyznaczonego wcześniej Eps-sąsiedztwa, dlatego zostanie on poddany przetwarzaniu jako kolejny. Proces od momentu wyznaczenia sąsiedztwa powtarza się.

Mimo, że zgodnie z rysunkiem 5.12c, w sąsiedztwie obiektu *p2* znajdują się dwa inne obiekty, odległość wewnętrzna zostanie ustalona tylko dla *p3*, ponieważ *p1* został już wcześniej oznaczony jako przeanalizowany. Zatem po przeliczeniu odległości wewnętrznej *p2*, jako następny zostaje wyznaczony obiekt *p3* (posiadający minimalną odległość osiągalną). W sąsiedztwie *p3* brak jest nieprzeanalizowanych obiektów, dlatego też algorytm wybiera kolejny ze zbioru danych (czyli *p4*). Obiekt *p4* nie spełnia warunku na minimalna liczbę obiektów w sąsiedztwie ($MinPts = 2$), w związku z czym, zarówno jego odległość osiągalna jak i wewnętrzna, pozostają niezdefiniowane. Jest on jedynie oznaczany jako poddany analizie (co zaprezentowano na rysunku 5.12d). Algorytm dokonuje zatem analizy i wyznaczenia odległości wewnętrznych dla pozostałych obiektów *p5* oraz *p6* (co zaznaczono na rysunku 5.12e, 5.12f). Należy jednak zaznaczyć, że efektem działania algorytmu OPTICS, jest określone uporządkowanie obiektów (zgodne z kolejnością ich przetwarzania, zapisywane po analizie każdego obiektu) oraz wyliczone odległości osiągalne i wewnętrzne.

Listing 5.3: Algorytm OPTICS

```
// ZbiorObiektow ma status NIEPRZEANALIZOWANY
OPTICS (ZbiorObiektow, Eps, MinPts, Uporzadkowanie)
Uporzadkowanie.open();
FOR i FROM 1 TO ZbiorObiektow.size DO
 Obiekt := ZbiorObiektow.get(i);
 IF NOT Obiekt.Processed THEN
  RozszerzUporzadkowanie(ZbiorObiektow, Obiekt, Eps, MinPts, Uporzadkowanie)
 END IF
 Uporzadkowanie.close();
END FOR
END; // OPTICS
```

Budowa algorytmu OPTICS została zaprezentowana na listingu 5.3. Podobnie jak w przypadku swojego gęstościowego pierwowzoru, potrzebuje on zdefiniowania promienia sąsiedztwa *Eps*, minimalnej liczby elementów jakie musi zawierać grupa (*MinPts*) oraz referencję do

struktury *Uporzadkowanie*, w której będzie zapisywana kolejność przetwarzania obiektów, wraz z odległościami wewnętrzną i osiągalną. Implementacja twórców algorytmu w [4] wykorzystuje w tym celu plik zlokalizowany na dysku twardym, jednakże jest to tylko sugestia i nic nie stoi na przeszkodzie, by te informacje przechowywać w pamięci RAM, tam gdzie to możliwe. Wpłynie to z całą pewnością korzystnie na czas działania samej techniki. Należy również nadmienić, że parametr *Eps* jest tutaj używany jako maksymalny próg sąsiedztwa, który będzie brany pod uwagę. Jego wartość powinna być relatywnie wysoka, ponieważ w późniejszym etapie podczas procesu grupowania, będzie można uzyskać w krótkim czasie, dowolny podział na grupy, uwzględniający promień sąsiedztwa od zera[6] do wartości *Eps*.

Naturalnie bazowy szkielet algorytmu jest bardzo podobny do listingu 5.1. Analizowany jest kolejno każdy, niezbadany dotychczas, obiekt ze zbioru danych wejściowych, poprzez wywołanie procedury *RozszerzUporzadkowanie*. Jej schemat został zaprezentowany na listingu 5.4.

Listing 5.4: Procedura RozszerzUporzadkowanie

```
RozszerzUporzadkowanie(ZbiorObiektow, Obiekt, Eps, MinPts, Uporzadowanie);
sasiedzi := ZbiorObiektow.neighbors(Obiekt, Eps);
Obiekt.Processed := TRUE;
Obiekt.odleglosc_osiagalna := NIEZDEFINIOWANA;
Obiekt.ustawOdlegloscWewnetrzna(sasiedzi, Eps, MinPts);
Uporzadowanie.write(Obiekt);
IF Obiekt.odleglosc_wewnetrzna <> NIEZDEFINIOWANA THEN
  OrderSeeds.update(sasiedzi, Obiekt);
  WHILE NOT OrderSeeds.empty() DO
    aktualnyObiekt := OrderSeeds.next();
    sasiedzi:=ZbiorObiektow.neighbors(aktualnyObiekt, Eps);
    aktualnyObiekt.Processed := TRUE;
    aktualnyObiekt.ustawOdlegloscWewnetrzna(sasiedzi, Eps, MinPts);
    Uporzadkowanie.write(aktualnyObiekt);
    IF aktualnyObiekt.odleglosc_wewnetrzna<>NIEZDEFIONIOWANA THEN
      OrderSeeds.update(sasiedzi, aktualnyObiekt);
    END IF
  END WHILE
END IF
END; // RozszerzUporzadkowanie
```

Początkowo, procedura *RozszerzUporzadkowanie* wyznacza sąsiedztwo dla obiektu przekazanego jako parametr z głównej pętli programu, ustawia jego *osiągalną odległość* na niezdefiniowaną i wylicza *odległość wewnętrzną*. W dalszej kolejności, badany jest warunek na obiekt wewnętrzny (w odniesieniu do zadanego *Eps*) i jeśli nie jest on spełniony, omawiana procedura przekazuje sterowanie do głównej pętli programu, która wybiera kolejny obiekt do analizy. W przeciwnym przypadku, odnajdywane są wszystkie obiekty, będące bezpośrednio gęstościowo osiągalne z aktualnie analizowanego i dodawane są one do listy *OrderSeeds*. Następnie, wszystkie elementy listy *OrderSeeds* są sortowane malejąco, na podstawie ich odległości osiągalnej do najbliższego obiektu wewnętrznego[7]. Proces ten powtarzany jest iteracyjnie i dla każdego

[6] Przy założeniu, że zero to najniższa, możliwa do określenia wartość promienia sąsiedztwa.

[7] Procedura sortowania implementowana jest zazwyczaj za pomocą kolejek priorytetowych, dlatego terminy te będą używane zamiennie.

obiektu (poczynając od tych z najmniejszą odległością osiągalną) z listy *OrderSeeds* wyznaczane jest sąsiedztwo i odległość wewnętrzna. Następnie, identyfikator obiektu wraz z wartościami odległości osiągalnej i wewnętrznej jest dopisywany do struktury *Uporzadkowanie*. Jeżeli analizowany kolejno obiekt zostałby także zaklasyfikowany jako wewnętrzny, dopisywany jest on do listy *OrderSeeds*, a cały proces jest powtarzamy do momentu, aż nie zostanie przeanalizowany każdy obiekt ze zbioru danych.

Listing 5.5: Metoda Update()

```
OrderSeeds::update(sasiedzi, ObiektCentralny);
c_dist := ObiektCentralny.odleglosc_wewnetrzna;
FORALL Obiekt FROM sasiedzi DO
  IF NOT Obiekt.Processed THEN
    new_r_dist:=max(c_dist,ObiektCentralny.dist(Obiekt));
  IF Obiekt.odleglosc_osiagalna=NIEZDEFINIOWANA THEN
    Obiekt.odleglosc_osiagalna := new_r_dist;
    insert(Obiekt, new_r_dist);
  ELSE // Obiekt jest już na liście OrderSeeds
    IF new_r_dist<Obiekt.odleglosc_osiagalna THEN
      Obiekt.odleglosc_osiagalna := new_r_dist;
      decrease(Obiekt, new_r_dist);
    END IF
  END IF
  END IF
END FORALL
END; // OrderSeeds::update
```

Za aktualizację listy *OrderSeeds* oraz wyznaczenie odległości osiągalnych odpowiada procedura *update(sasiedzi, Obiekt)*, przedstawiona na listingu 5.5. Obiekty, które nie znajdują się jeszcze w kolejce priorytetowej *OrderSeeds*, są do niej dopisywane na podstawie wyliczonej odległości osiągalnej (względem obiektu centralnego). Jeżeli dany obiekt znajdował się już w kolejce i jego zaktualizowana wartość odległości osiągalnej jest mniejsza niż poprzednia, jest on przestawiany na wyższe miejsce kolejki.

Złożoność obliczeniowa i zajętość pamięci

Budowa algorytmu OPTICS wykazuje wysoki stopień podobieństwa do techniki DBSCAN co warunkuje, że czas działania OPTICS (a w konsekwencji również jego złożoność obliczeniowa) tylko nieznacznie różni się od czasu działania pierwowzoru. Czynnikiem, który ma największe znaczenie przy określaniu złożoności obliczeniowej, jest wyznaczanie sąsiedztwa danego obiektu, ponieważ jest to operacja wykonywana dla każdego elementu zbioru danych. Zatem w przypadku pesymistycznym, bez uwzględnienia żadnych struktur indeksujących, wspomniana złożoność jest rzędu $O(n^2)$. Potwierdza to również analiza eksperymentalna, przeprowadzona w [4], w wyniku której stwierdzono, że czas działania algorytmu OPTICS jest większy 1.6 razy w stosunku do techniki DBSCAN. Można ten czas zminimalizować, poprzez zastosowanie R*-drzew jako struktury do przechowywania danych wejściowych. Wówczas średnia złożoność obliczeniowa maleje do $O(n \cdot \log n)$.

Pod kątem zajętości pamięci, algorytm przechowuje jedynie niezbędne minimum informacji, takich jak identyfikatory obiektów (ustawionych w odpowiedniej kolejności), jak również wartości odległości wewnętrznych i osiągalnych. Średnia zajętość pamięci jest wprost proporcjonalna do liczby przechowywanych obiektów i jest rzędu $O(n)$.

OPTICS jako narzędzie grupowania danych

Technika OPTICS może w prosty sposób zostać przystosowana do generowania podziału na skupienia. Schemat algorytmu pomijał jedynie przydział określonych identyfikatorów grup dla analizowanych obiektów. Takie postępowanie wynika z faktu, że mając wygenerowane uporządkowanie zbioru danych, można na jego podstawie, wydobyć grupy dla dowolnej wartości parametru sąsiedztwa (mniejszej od ustalonego wcześniej maksymalnego *Eps*), poprzez analizę odległości osiągalnych i wewnętrznych. Listing 5.6 przedstawia algorytm *WydobadzSkupienia* realizujący wspomniane zadanie.

Listing 5.6: Algorytm WydobadzSkupienia

```
WydobadzSkupienia(UporzadkowaneObiekty, Eps_i, MinPts)
//Eps_i ⩽ Eps dla UporzadkowaneObiekty
SkupienieId := SZUM;
FOR i FROM 1 TO UporzadkowaneObiekty.size DO
  Obiekt := UporzadkowaneObiekty.get(i);
  IF Obiekt.odleglosc_osiagalna > Eps_i THEN
    // Założenie: NIEOKREŚLONA > Eps_i
    IF Obiekt.odleglosc_wewnetrzna ⩽ Eps_i THEN
      SkupienieId := nextId(SkupienieId);
      Obiekt.skupienieId := SkupienieId;
    ELSE
    Obiekt.skupienieId := SZUM;
  END IF
  ELSE // Obiekt.odleglosc_osiagalna ⩽ Eps_i
  Obiekt.skupienieId := SkupienieId;
  END IF
END FOR
END; // WydobadzSkupienia
```

Procedura *WydobadzSkupienia* analizuje kolejno każdy obiekt, zgodnie z uzyskanym uporządkowaniem. Na początku testowane jest, czy odległość osiągalna aktualnie analizowanego obiektu, jest większa niż wartość promienia sąsiedztwa ($Eps_i \leqslant Eps$), dla jakiego ma zostać utworzony podział na grupy. Wówczas aktualny obiekt, nie jest gęstościowo-osiągalny z żadnego innego obiektu umieszczonego przed nim, w stworzonym uporządkowaniu. Jeżeli natomiast osiągalna odległość jest większa niż Eps_i, sprawdzany jest warunek na obiekt wewnętrzny, a po jego spełnieniu tworzone jest nowe skupienie. W przeciwnym przypadku, dany obiekt jest klasyfikowany jako szum informacyjny. Jeżeli natomiast odległość osiągalna aktualnego obiektu jest mniejsza bądź równa Eps_i, jest on przypisywany do aktualnego skupienia. Intuicyjnie rzecz biorąc, jest on wtedy gęstościowo-osiągalny z poprzedzającego go w uporządkowaniu obiektu wewnętrznego.

W odróżnieniu od algorytmu DBSCAN, przedstawiony sposób umożliwia szybkie wygenerowanie dowolnego podziału, dla ustalonego *MinPts* i promienia sąsiedztwa z przedziału $[0, Eps]$. Cała procedura opiera się bowiem na pojedynczym przeglądzie zupełnym uporządkowanej struktury obiektów – rezultatu działania techniki OPTICS. Samo generowanie wspomnianej struktury odbywa się jednokrotnie dla zbioru danych (przy $MinPts = const$), dlatego też, z punktu widzenia procesu grupowania, można ten czas pominąć. Umożliwia to, poddanie ocenie uzyskanych podziałów na skupienia i wybór jedynie najlepszego jakościowo.

Zalety i wady metody OPTICS

Algorytm OPTICS eliminuje istotną wadę swojego pierwowzoru (algorytmu DBSCAN), mianowicie trudność w znajdywaniu skupień, charakteryzujących się występowaniem hierarchii (mniejsze skupienia zawarte w większym) i różnym stopniem zagęszczenia. Mimo, że sam algorytm OPTICS w sposób bezpośredni nie generuje skupień, a jedynie określone uporządkowanie obiektów, w bardzo prosty sposób można go do tego celu przystosować. Ponadto, uzyskane w wyniku działania informacje, mogą być wykorzystane do wizualizacji wewnętrznej struktury danych, a w konsekwencji łatwiejszej analizy powiązań między nimi [4]. Największą wadą tego algorytmu jest brak przetestowanej heurystyki do wyznaczania wartości parametru *MinPts*.

5.4 Podsumowanie

Celem niniejszego rozdziału było omówienie gęstościowych algorytmów analizy skupień oraz potwierdzenie ich użyteczności w kontekście grupowania rzeczywistych zbiorów danych złożonych. Przedstawiono proste przykłady działania algorytmów, jak również ich budowę w postaci pseudokodu. Szczególnym aspektem analizy, było porównanie algorytmu DBSCAN z algorytmem OPTICS. Zestawienie to wykazało, że analizowane algorytmy posiadają między sobą wiele podobieństw, mimo że mogą być wykorzystywane w zupełnie różnych celach – algorytm OPTICS można bowiem zastosować wyłącznie jako narzędzie wizualizacji[8] struktury danych, bez generowania skupień. Warto również wspomnieć, że istnieją inkrementalne wersje omówionych algorytmów (szczegółowo przedstawione w [37, 2]), dzięki czemu można je zastosować w odniesieniu do często aktualizowanych baz danych. Taka sytuacja jednak nie ma miejsca w przypadku analizowanych zbiorów złożonych, dlatego też nie zostały one przeanalizowane dogłębnie w niniejszym rozdziale.

[8] Zastosowanie algorytmu OPTICS w celach wizualizacyjnych, zostanie omówione w rozdziale 6.

Rozdział 6

Graficzne metody reprezentacji skupień

Nieustanny rozwój techniki oraz rosnące możliwości sprzętu komputerowego, umożliwiają przechowywanie bardzo dużej liczby danych, we wszelkiego typu bazach i repozytoriach. Dane te najczęściej zbierane są w sposób automatyczny, wykorzystując szereg czujników lub systemów monitorujących. Są one gromadzone, ponieważ zakłada się, że mogą być źródłem nieznanych, potencjalnie użytecznych wzorców, korelacji i trendów. Niestety, skutkuje to również pojawieniem się tzw. problemu nadmiaru informacji (*ang. information overload problem*[1]) [31]. Odkryte (poprzez zastosowanie technik eksploracji danych) wzorce, wyrażone w postaci modelu analitycznego, mogą posiadać skomplikowaną strukturę, przez co są trudne w dalszej analizie i interpretacji, a w konsekwencji nie dostarczają nowej wiedzy oraz mogą doprowadzić do podjęcia błędnych decyzji. Dlatego coraz to częściej używane są narzędzia wizualizacyjne, które wykorzystując naturalne predyspozycje człowieka do przetwarzania graficznych źródeł informacji, prezentują odkryte zależności w bardziej przystępny i zrozumiały dla odbiorcy sposób. Dobrym przykładem jest prezentacja prognozy pogody w postaci ikon rozmieszczonych na mapie danego obszaru (symbolizujących zachmurzenie czy opady). Pozwala to bardzo szybko przyswoić i zrozumieć relacje czy trendy klimatyczne nawet na przestrzeni kilku dni, co byłoby znacząco utrudnione, gdyby te same informacje zostały przedstawione w postaci szeregu tabel czy statystyk. Podobny przykład, odnośnie przedstawiania warunków atmosferycznych dla trzech różnych miast, został przedstawiony w formie tabelarycznej i graficznej, w pracy [3].

Niniejszy rozdział odnosi się do problemu, w jaki sposób wizualizacja danych może funkcjonować jako efektywne i autonomiczne narzędzie analizy danych, jak również służyć jako technika łącząca wiedzę dziedzinową i zdolności kognitywne człowieka, w procesie odkrywania wiedzy. Omawia proces graficznej analizy eksploracyjnej (*ang. visual data mining*) oraz dokonuje porównania najpopularniejszych technik reprezentacji skupień, spotykanych w literaturze przedmiotu, z przyjętą przez autora koncepcją, opartą na algorytmie generowania tzw. map prostokątów (*ang. treemaps*[2]).

[1] Problem nadmiaru informacji został szczegółowo omówiony m.in. w [63].

[2] Ze względu na ograniczoną liczbę polskich pozycji literaturowych, autor dokonał przetłumaczenia angielskich nazw technik wizualizacji, tam gdzie to było uzasadnione i możliwe.

6.1 Motywacja do wykorzystania technik wizualizacji danych

W literaturze przedmiotu [32, 61] spotyka się coraz częściej pojęcie graficznej analizy eksploracyjnej[3] (*ang. visual data mining*), która bazuje na wrodzonych zdolnościach kognitywnych człowieka. Jej podstawą jest wykorzystanie technik wizualizacji, w celu analizy i eksploracji posiadanych danych, poprzez zilustrowanie zawartej niejawnie w nich wiedzy. Mimo iż techniki wizualizacji danych wykorzystywane są od wielu lat, to jednak sam termin *graficzna analiza eksploracyjna* został po raz pierwszy użyty dopiero nieco ponad dekadę temu, by zaakcentować fakt, że jest to bardziej złożony proces.

Motywacją do wyróżnienia tego pojęcia, były dyskusje badaczy odnośnie idealnego narzędzia do wydobywania wiedzy. Mimo, że sukcesywnie wdrażane są coraz to bardziej zaawansowane narzędzia automatycznej analizy danych, wciąż aktualny wydaje się problem zrozumienia i interpretacji uzyskanych wyników. W pełni autonomiczne podejścia przeszukiwania czy ekstrakcji wiedzy działają niezawodnie tylko dla dobrze zdefiniowanych i określonych problemów [31]. Ponadto, zwykle nie posiadają one możliwości objaśniania sposobu dojścia do określonych wzorców czy zależności. Jest to szczególnie istotne w przypadku, gdy na podstawie uzyskanych wyników analiz mają zostać podjęte konkretne decyzje (np. biznesowe bądź medyczne). Dlatego też, Ankrest w [3] stwierdził, że dostępne techniki analizy są nieefektywne, ponieważ nie uwzględniają one bezpośredniego udziału użytkownika postrzeganego przede wszystkim jako analityka, który chce mieć wgląd bądź lepiej poznać zależności występujące w analizowanym zbiorze danych. W [61] przytoczono również opinię, że najlepsze narzędzie ekstrakcji danych powinno być w znacznym stopniu interaktywne.

Wykorzystanie technik wizualizacji, w procesie odkrywania wiedzy, doprowadziło do rozbieżności w rozumieniu przez naukowców pojęć związanych z tą tematyką, takich jak: analiza wizualna (*ang. visual analytics*), wizualizacja informacji (*ang. information visualization*), czy graficzna analiza eksploracyjna (*ang. visual data mining*). W [65] Thomas i Cook definiują analizę wizualną jako dziedzinę nauki, zajmującą się wnioskowaniem analitycznym, wspomaganym przez wykorzystanie interaktywnych technik wizualizacji. Ponadto w pracy [31] przytoczono szerszą definicję tego pojęcia, jako połączenie automatycznych technik analizy z interaktywną wizualizacją, dla efektywnego pojmowania, wnioskowania i podejmowania decyzji, na podstawie dużych zbiorów danych złożonych. Celem tej dziedziny jest zatem generowanie wiedzy (w postaci wzorców, podsumowań i trendów) z dużych zbiorów danych, potwierdzenie znanych zależności i odkrycie tych niejawnych, jak również dostarczenie zrozumiałych i prawidłowych oszacowań, wspomagając ewentualny proces decyzyjny. Natomiast wizualizacja informacji została w [54] określona jako wykorzystanie (wspomaganej komputerowo) interaktywnej, graficznej reprezentacji danych, by dokonać ich abstrakcji, celem pobudzenia percepcji. Oba pojęcia są ze sobą zgodne w aspekcie wykorzystywania technik wizualizacji, jak również interakcji z użytkownikiem. Jednakże wizualizacja informacji może służyć np. wyłącznie ich prezentacji w lepiej przyswajalnej formie, natomiast analiza wizualna silnie wykorzystuje metody analizy danych – jest to zatem połączenie wizualizacji, procesów poznawczych człowieka i metod uczenia maszynowego. Pojęcie graficznej analizy eksploracyjnej eksponuje przede wszystkim

[3] Szczegółowa definicja pojęcia graficznej analizy eksploracyjnej znajduje się w dalszej części niniejszego rozdziału.

wykorzystanie technik eksploracji danych w procesie odkrywania nowej i użytecznej wiedzy. W [32] stwierdzono, że celem graficznej eksploracji jest przedstawienie użytkownikowi struktury danych, aby na tej podstawie móc wyciągać wnioski, dotyczące ukrytych w nich zależności. Zatem techniki wizualizacji są traktowane jako zupełnie autonomiczne narzędzie, które pomaga w zrozumieniu rozkładu czy struktury analizowanego zbioru. Ankerst [3] natomiast, skupiał się na relacji między wizualizacją a całym procesem odkrywania wiedzy, definiując graficzną analizę eksploracyjną jako "krok w procesie odkrywania wiedzy, który wykorzystuje graficzną prezentację jako kanał komunikacyjny między systemem komputerowym a użytkownikiem, celem identyfikacji nieznanych i zrozumiałych wzorców". Należy zatem zaznaczyć, że proces odkrywania wiedzy jest silnie zależny od posiadanej wiedzy dziedzinowej i powinien być sterowany przez użytkownika (najlepiej będącego ekspertem z danej dziedziny). Ze względu na istotne różnice między przytoczonymi definicjami, na potrzeby niniejszej pracy postanowiono określić graficzną analizę eksploracyjną jako proces interakcji i wnioskowania analitycznego z wykorzystaniem wizualnej reprezentacji abstrakcyjnego zestawu danych, który prowadzi do odkrycia nieznanych i potencjalnie użytecznych informacji lub wiedzy (najczęściej zakodowanej w formie wzorców czy trendów).

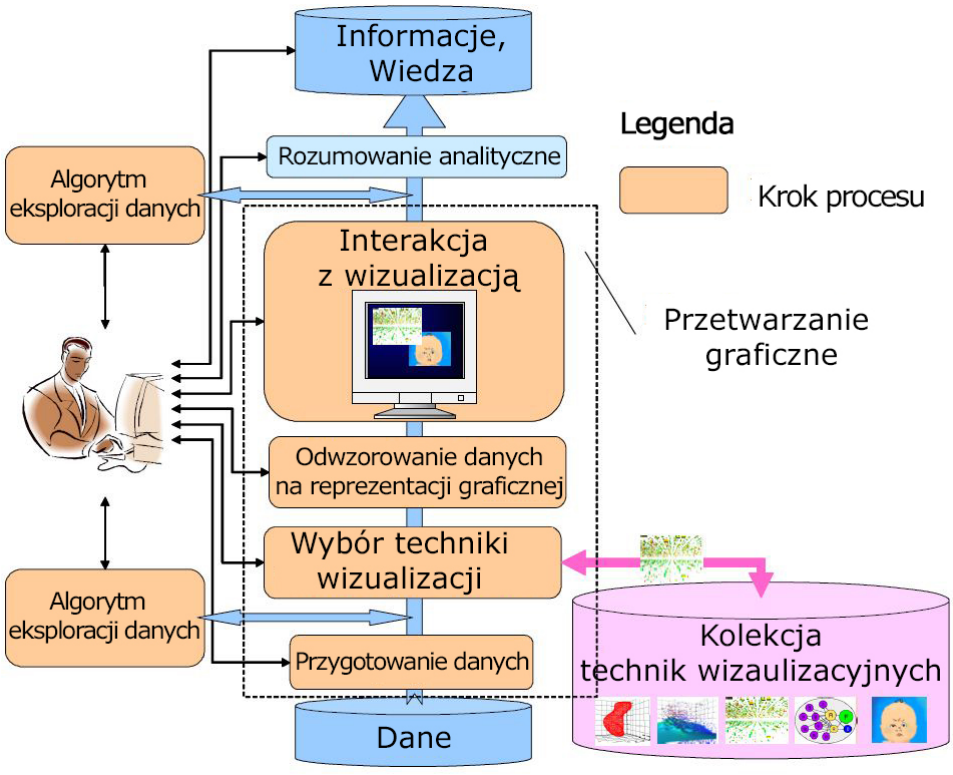
6.2 Proces graficznej analizy eksploracyjnej

Aby proces wydobywania wiedzy z wykorzystaniem technik wizualizacyjnych był efektywny, użytkownik powinien mieć możliwość wglądu do każdego z jego etapów i dostosowania ich do potrzeb aktualnie realizowanego zadania. Podstawową ideą graficznej analizy eksploracyjnej jest zaprezentowanie użytkownikowi zbioru danych (lub jego abstrakcji) w formie graficznej, dzięki czemu uzyskuje on wgląd w jego strukturę czy interesujące zależności. Jest to szczególnie przydatne, gdy niewiele wiadomo na temat przetwarzanego zbioru lub cele badawcze są bardzo ogólne. Ponieważ użytkownik jest bezpośrednio zaangażowany w proces eksploracji, zmiana bądź dostosowanie aktualnych planów badawczych (np. poprzez ich uszczegółowienie) odbywa się na bieżąco w miarę potrzeb. W pracy [18] wyszczególniono następujące, możliwe do realizacji, cele graficznej analizy eksploracyjnej:

- ocena kompletności i prawidłowości zebranych danych,
- detekcja wyjątków, anomalii oraz obserwacji odstających,
- rozpoznanie i zrozumienie trendów, tendencji, podobieństw czy symetrii,
- wyodrębnienie skupień,
- porównanie różnic między grupami,
- weryfikacja założeń dotyczących rozkładu danych,
- pomiar postępów i obserwacja procesu w czasie,
- predykcja oraz ocena potencjalnych, przyszłych trendów.

Autor niniejszej pracy planuje wykorzystać techniki wizualizacyjne jako metodę wykrywania odchyleń (wykorzystując wykresy pudełkowe bądź analizy częstości), celem zidentyfikowania np. wadliwych bądź niewłaściwie skonfigurowanych urządzeń sieciowych. Istotnym zadaniem jest także reprezentacja graficzna wyodrębnionych skupień (powstałych przez zastosowanie algorytmu gęstościowego), dzięki czemu możliwe będzie skoncentrowanie uwagi na interesujących wzorcach (grupach) i interpretacja odkrytych powiązań. Przy analizie badawczej dużych zbiorów danych złożonych, potencjalnie otrzymuje się grupy liczone w tysiącach, co utrudnia ich analizę bazując tylko i wyłącznie na narzędziach statystycznych.

Proces graficznej eksploracji danych[4] może być postrzegany również jako formułowanie hipotez: wizualizacja danych na różnych poziomach szczegółowości może prowadzić do formułowania nowych teorii i ich potwierdzenia. Prócz wykorzystywania wiedzy dziedzinowej i wrodzonych zdolności poznawczych człowieka, omawiane podejście ma następujące zalety w stosunku do automatycznych metod statystycznych czy technik uczenia maszynowego: proces zrozumienia i interakcji z wizualizacją jest zazwyczaj intuicyjny oraz lepiej radzi sobie w przypadku silnie zaszumionych czy niejednorodnych danych [32].



Rysunek 6.1: Etapy procesu graficznej analizy eksploracyjnej.
Źródło: [3]

Szczegółowy przebieg graficznej analizy eksploracyjnej został przedstawiony na rysunku 6.1. Użytkownik może ingerować w każdy etap przedstawionego procesu, począwszy od przygoto-

[4]Pojęcia graficznej eksploracji danych i graficznej analizy eksploracyjnej wykorzystywane są zamiennie i odnoszą się do tego samego procesu.

wania danych źródłowych (np. poprzez wybór atrybutów najbardziej istotnych z punktu widzenia realizowanego problemu odkrywania wiedzy) czy wyboru najlepszej techniki wizualizacji danych (dostosowanej do specyfiki analizowanego zbioru), aż po dobór parametrów wygenerowanej wizualizacji i analizę uzyskanego rezultatu. Przykładowo użytkownik może powiększyć wyłącznie interesujący obszar stworzonej wizualizacji i na nim skupić swoje dalsze działania. Ma to bardzo często miejsce w kontekście analizy złożonych, wielowymiarowych zbiorów. Ponadto w takich przypadkach można również zastosować różnorakie algorytmy eksploracji danych – zarówno przed wygenerowaniem jakiejkolwiek wizualizacji, jak również po wstępnej jej analizie. W pierwszym przypadku, wynik działania takich algorytmów może być traktowany jako nowy zbiór wejściowy poddawany wizualizacji. Przykładowo można już na początku procesu eksploracji wygenerować reguły asocjacyjne, które będą wizualizowane na wykresach słupkowych. W dalszym etapie taką wizualizację można poddać grupowaniu, by zlokalizować graficznie skupienia, podobnych pod względem semantycznym reguł i dokonać dalszych analiz. Techniki eksploracji danych można również zastosować w końcowym etapie omawianego procesu, celem poprawy czytelności samej wizualizacji np. wykorzystując grupowanie do agregacji podobnych przypadków na wykresie, zmniejszając jego objętość. Istotny w tym procesie jest również dobór właściwej techniki oddziaływania oraz metody wizualizacji danych.

6.3 Reprezentacja skupień

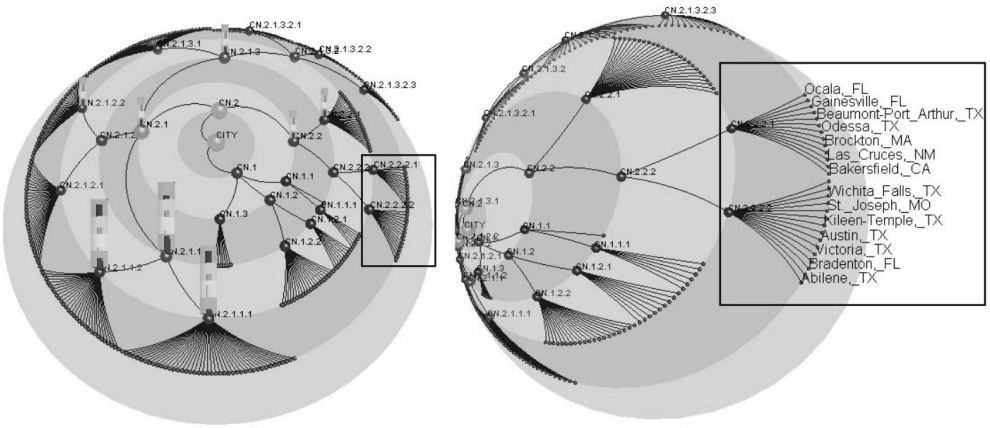
Jak umotywowano we wcześniejszych rozdziałach, jako podstawową technikę eksploracji danych wykorzystywaną w pracy, zdecydowano się wybrać analizę skupień. Będzie ona wykorzystywana przez autora rozprawy zarówno jako sposób generowania nowych i potencjalnie użytecznych wzorców, które będą wizualizowane w celu ich łatwiejszego zrozumienia, jak również jako podejście do kompresji prezentowanych danych. Planuje się dokonanie dwuetapowego grupowania (w pierwszej fazie na całym zbiorze danych, w drugiej ograniczonego wyłącznie do zbioru reprezentantów stworzonych w pierwszym kroku grup) oraz przedstawienie skupień w formie map prostokątów [5] i wizualizację jedynie interesującego w danej chwili poziomu uzyskanej hierarchii. Potencjalnie ułatwi to zrozumienie powstałych zależności i ich analizę, która jest utrudniona, gdyż autor spodziewa się uzyskania kilku tysięcy skupień po pierwszym grupowaniu (dla rzeczywistych zbiorów danych badanych w rozprawie). Dlatego też niniejsza sekcja będzie poświęcona przeglądowi metod ich graficznej reprezentacji, w odniesieniu do przetwarzania danych złożonych[6].

Niewątpliwie najczęściej spotykaną formą reprezentacji skupień w literaturze jest dendrogram lub odwzorowanie w postaci diagramu Woronoja. Dendrogram[7] jest diagramem drzewiastym, ilustrującym związki pomiędzy wybranymi elementami (obiektami lub grupami) na podstawie przyjętego kryterium (podobieństwa). Największą zaletą tej techniki jest możliwość bezpośredniego reprezentowania hierarchii skupień jak również łatwość w interpretacji powsta-

[5] Technika map prostokątów zostanie omówiona w dalszej części niniejszego rozdziału.

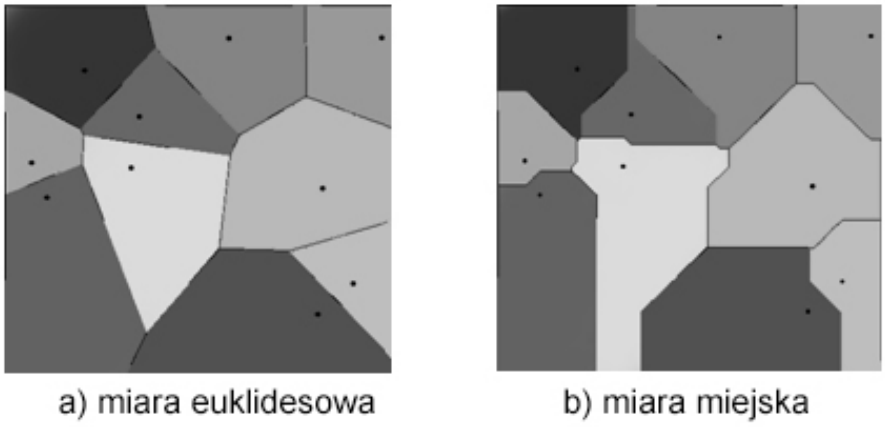
[6] Mimo, że celem niniejszego rozdziału jest przedstawienie graficznych metod reprezentacji skupień, autor dokonał w pracy [49] również analizy i porównania najpopularniejszych w literaturze przedmiotu metod reprezentacji danych.

[7] Bardziej szczegółowa definicja tego pojęcia znajduje się m.in. w [64].



Rysunek 6.2: Zastosowanie techniki Magic-Eye-View do reprezentacji struktur hierarchicznych. Źródło: [17]

łych związków. Ponadto dzięki tej formie wizualizacji, można w relatywnie łatwy sposób objaśnić i przeanalizować, dlaczego wybrane obiekty zostały połączone w jedną grupę. Niestety czytelność tej techniki ulega pogorszeniu wraz ze wzrostem liczby przedstawianych elementów (co utrudnia jej zastosowanie w kontekście analizy dużych, złożonych zbiorów). W celu zmniejszenia tego problemu wykorzystuje się przybliżanie wybranych fragmentów drzewa lub odwzorowanie struktury hierarchicznej na płaszczyznę półkuli. Wówczas, fragment dendrogramu będący w obszarze aktualnego zainteresowania użytkownika zostaje powiększony, a pozostała jego część jest przesuwana na obrzeże półkuli. Taka sytuacja została zaprezentowana na rysunku 6.2, wykorzystując tzw. metodę Magic-Eye-View [17]. Mimo prezentacji całej struktury skupień, technika ta wprowadza znaczące zniekształcenia, co może prowadzić do sformułowania błędnych wniosków podczas analiz.



Rysunek 6.3: Porównanie diagramów Woronoja dla różnych miar odległości. Źródło: http://en.wikipedia.org/wiki/Voronoi_diagram

Diagram Woronoja[8] wykorzystywany jest najczęściej w sytuacji przedstawienia płaskiego

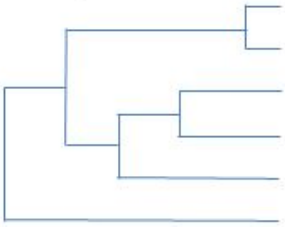
[8]Formalna definicja i sposób tworzenia diagramów Woronoja zostały przedstawione w [40].

podziału (np. wygenerowanego z użyciem algorytmu k-średnich). Zaletą tej techniki jest wykorzystanie całego obszaru roboczego ekranu (dzięki czemu większa liczba skupień może zostać przedstawiona, niż poprzez użycie dendrogramu), jednakże w swoim klasycznym ujęciu nie umożliwia ona wizualizowania hierarchii (chyba, że każdy jej poziom zostanie przedstawiony jako osobny diagram). Największą wadę tej techniki stanowi fakt, że rozkład i kształt komórek[9] na diagramie jest silnie zależny od przyjętej miary podobieństwa obiektów. Taka sytuacja, dla sztucznie wygenerowanego zbioru danych podzielonego na 10 grup, została przedstawiona na rysunku 6.3.

a)

Badanie	Pomiar 1	Pomiar 2	Pomiar 3
BD-1	2,01	3,22	5,58
BD-2	2,93	4,36	6,01
BD-3	2,64	5,02	7,19
BD-4	2,37	3,48	8,31
BD-5	2,21	3,12	5,38
BD-6	2,44	3,84	6,47

b)



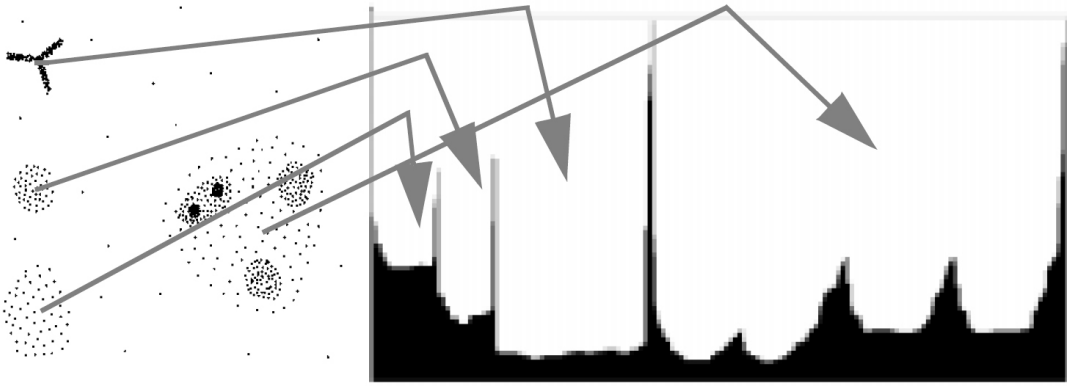
Badanie	Pomiar 1	Pomiar 2	Pomiar 3
BD-5	2,21	3,12	5,38
BD-1	2,01	3,22	5,58
BD-6	2,44	3,84	6,47
BD-2	2,93	4,36	6,01
BD-3	2,64	5,02	7,19
BD-4	2,37	3,48	8,31

Rysunek 6.4: Reprezentacja skupień za pomocą mapy ciepła.
Źródło: Opracowanie własne

Kolejną techniką, która nie jest wprost dedykowana do przedstawiania skupień, jednakże może zostać bardzo łatwo do tego przystosowana jest wykorzystanie tzw. map ciepła (*ang. heatmap*). Ilustruje ona dane w formie macierzy, gdzie wiersze reprezentują poszczególne obiekty, natomiast kolumny ich atrybuty oraz dodatkowo każda wartość (na przecięciu wiersza i kolumny) ma przypisany kolor zgodny ze wcześniej ustalonym schematem. Przedstawiono to na rysunku 6.4a – ciemnozielony kolor oznacza niskie wartości, natomiast ciemnoczerwony wysokie. Jeżeli obiekty danych zostaną uporządkowane na podstawie wyniku grupowania hierarchicznego i zaznaczone będą (jak na rysynku 6.4b pogrubionymi liniami) granice skupień, to można tę technikę zastosować nie tylko do reprezentacji grup, ale również jako prostą metodę wizualnej oceny dlaczego obiekty zostały w ten sposób pogrupowane. Czytelność tej techniki zależy od przyjętej liczby kolorów i ulega pogorszeniu wraz ze wzrostem przetwarzanej liczby cech oraz obiektów [36].

Metodami graficznego przedstawienia struktury zbioru danych i skupień zajmowali się również twórcy algorytmu gęstościowego OPTICS (omówionego w sekcji 5.3). Zaproponowali oni

[9] Obszarów reprezentujących skupienie.



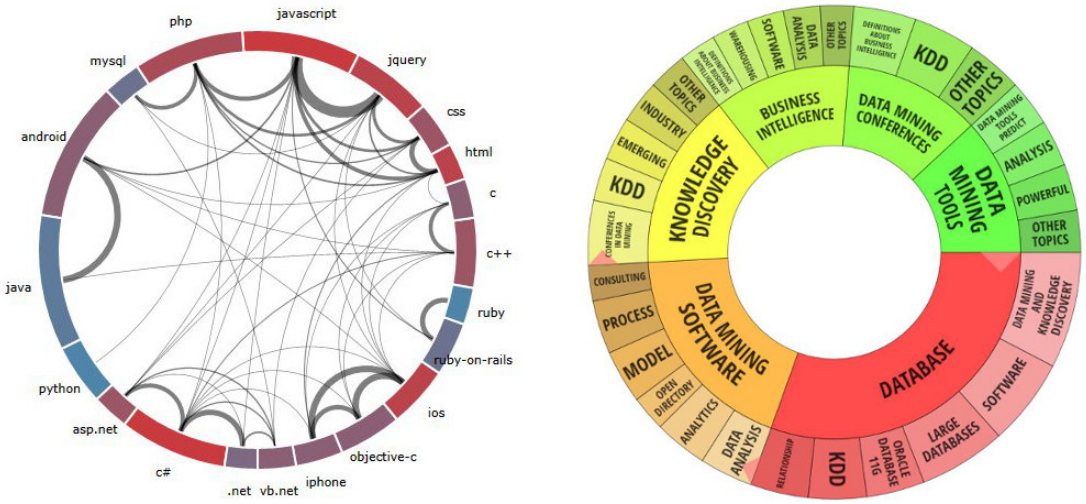
Rysunek 6.5: Zagnieżdżone skupienia na wykresie osiągalności.
Źródło: [4]

stworzenie tzw. wykresu osiągalności (*ang. reachability plot*) [4]. Jest to dwuwymiarowy wykres słupkowy funkcji odległości osiągalnej dla każdego obiektu ze zbioru danych, zgodnie z wygenerowanym przez algorytm OPTICS porządkiem. Rysunek 6.5 przedstawia wykres osiągalności dla przykładowego zestawu sztucznie wygenerowanych danych. Na osi rzędnych zaznaczana jest wartość odległości osiągalnej, natomiast na osi odciętych zaznaczany jest najczęściej numer bądź identyfikator danego obiektu. Skupienia na wykresie reprezentowane są przez doliny. Im dana dolina jest węższa, tym mniej obiektów wchodzi w skład danego skupienia. Natomiast im mniejsza jest wartość osiągalnej odległości, tym skupienie jest bardziej zagęszczone (spójne). Zaletą wizualizacji za pomocą wykresu osiągalności jest możliwość identyfikacji zagnieżdżonych skupień. Hierarchia skupień (rys. 6.5) jest widoczna jako szereg małych dolin zawartych w jednej głębszej. Niestety jak zostanie udowodnione podczas analiz eksperymentalnych, dla rzeczywistych zbiorów danych złożonych, identyfikacja hierarchii skupień z wykorzystaniem opisywanej metody jest zdecydowanie trudniejsza, przez co technika ta nie może być w każdym przypadku bezpośrednio wykorzystana[10].

Powszechnie znany jest sposób wizualizacji danych za pomocą wykresów kołowych. Można je również wykorzystać jako narzędzie reprezentowania hierarchii skupień, ale także jako ilustrację powiązań między grupami. W przypadku pierwszego zadania, koło dzielone jest na tyle wycinków, ile grup znajduje się na wizualizowanym poziomie hierarchii. Szerokość każdego wycinka najczęściej związana jest z wielkością danego skupienia – im grupa zawiera więcej obiektów, tym jego obszar jest większy. Następnie poszczególne wycinki można poddać proporcjonalnie dalszej dekompozycji, zgodnie ze strukturą skupień, obrazując dalsze poziomy hierarchii. Koncepcja ta została już wykorzystana przez twórców wyszukiwarki Carrot2[11], która to grupuje tematycznie strony internetowe i za pomocą biblioteki Search Circles dokonuje wizualizacji wygenerowanej hierarchii. Rezultat dla hasła *data mining* przedstawia rysunek 6.6b. Z przedstawionego rysunku wyraźnie wynika, że znalezione dokumenty można podzielić na: opisujące bazy danych, odkrywanie wiedzy, narzędzia eksploracji danych, oprogramowanie czy

[10] Bardziej szczegółowe omówienie wykresu osiągalności zostało dokonane przez autora w [71].

[11] Szczegółowe informacje na temat wyszukiwarki Carrot2 oraz bibliotek do wizualizacji danych znajdują się pod adresem `http://carrotsearch.com/index`.



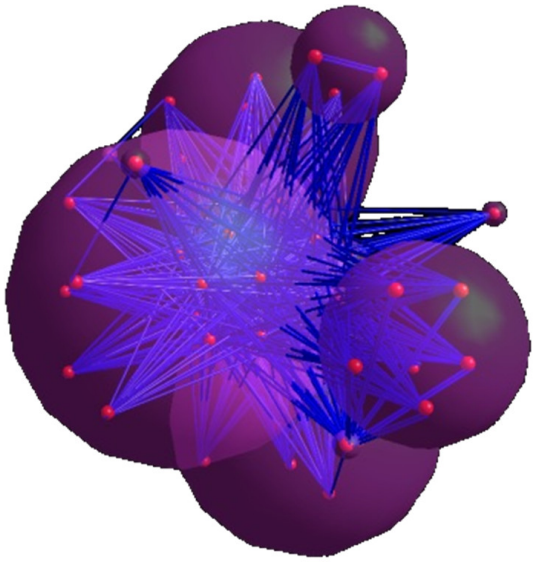
a) wizualizacja powiązań b) reprezentowanie hierarchii

Rysunek 6.6: Wykresy kołowe jako reprezentacja powiązań i hierarchii skupień.
Źródło: http://www.codeproject.com/Articles/342715/Plotting-Circular-Relationship-Graphs-with-Silverl, http://download.carrotsearch.com/circles/demo/demos/tweakpanel.html

konferencje poświęcone tym zagadnieniom. Jest to struktura hierarchiczna, ponieważ konferencje można dalej tematycznie dekomponować na poświęcone procesowi odkrywania wiedzy w bazach danych, tematyce analityki biznesowej (*ang. business intelligence*) i pozostałym zagadnieniom. Przedstawiona metoda wizualizacji niestety nie wykorzystuje najlepiej dostępnej przestrzeni roboczej, przez co prezentowana liczba skupień musi być dość ograniczona, by zachować wysoką czytelność.

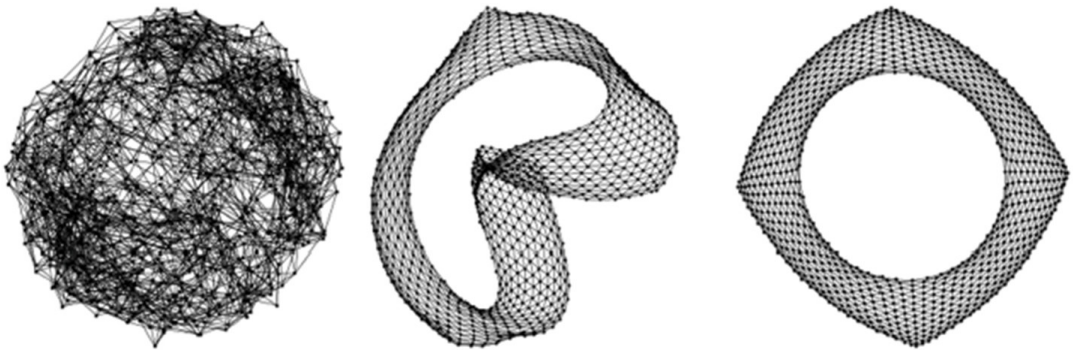
W celu wizualizacji powiązań za pomocą wykresów kołowych, między poszczególnymi wycinkami rysowane są łuki, których grubość oznacza siłę takiego powiązania. W przytoczonym na rysunku 6.6a przykładzie, pogrupowano zapytania użytkowników forum internetowego odnośnie problemów programistycznych. Widać, że użytkownicy najczęściej pytają o informacje związane z językiem java i systemem android (ponieważ tak opisane wycinki na rysunku 6.6a są najszersze, a ich wielkość jest tożsama z liczbą pytań na dany temat). Te dwa zagadnienia często występują wspólnie, przez co są ze sobą silnie powiązane (co zostało zobrazowane grubszą szarą linią). Warto zauważyć, że każdy wycinek koła jest kolorowany na podstawie liczby powiązań między pozostałymi. Zimne kolory (fioletowy czy niebieski) oznaczają, że nie da się zaobserwować zbyt dużego związku między danymi zagadnieniami a pozostałymi technologiami – przykładowo język python nie jest wykorzystywany w połączeniu z językiem c czy ruby. Najsilniej w badanym gronie zapytań powiązane są ze sobą technologie javascript i jquery — tj. pytania na ich temat, najczęściej występują wspólnie, co wynika z faktu, że jquery to biblioteka napisana w tym języku.

Również doskonale znane techniki grafowe mogą zostać wykorzystane w przedstawieniu i identyfikacji skupień. Jest to często wykorzystywane w analizie sieci społecznościowych – grafy reprezentują wówczas powiązania między poszczególnymi ich członkami. Jeżeli wykorzy-



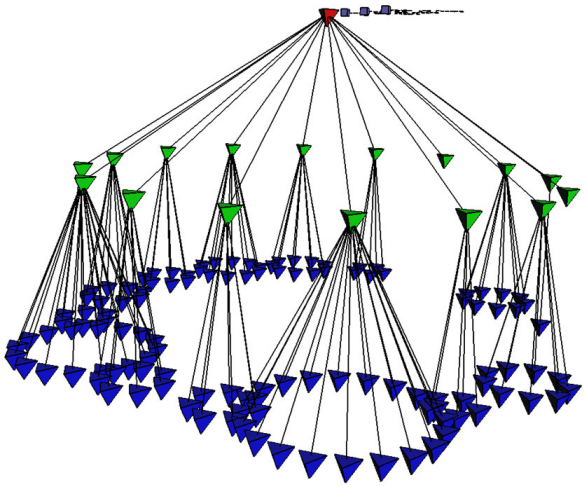
Rysunek 6.7: Techniki grafowe jako narzędzie reprezentowania powiązań.
Źródło: http://wilma.sourceforge.net/

stana zostanie przestrzeń trójwymiarowa, wówczas można na podstawie takich połączeń i przy zastosowaniu technik analizy skupień, zaznaczyć granice grup. Taka sytuacja (dla sztucznego zbioru danych) została zobrazowana na rysunku 6.7, gdzie kolorem fioletowym zaznaczono kuliste skupienia. Niestety takie podejście, choć wydaje się intuicyjnie poprawne, może zaburzyć naturalnie występujące tendencje w danych. Graf bowiem można przedstawić w przestrzeni trójwymiarowej na wiele różnych sposobów. Na rysunku 6.8 przedstawiono trzy sposoby prezentacji tego samego grafu (składającego się z 936 węzłów). Jeżeli skupienia wykrywane byłyby na podstawie reprezentacji grafu przedstawionego po stronie lewej, to można się doszukać dwóch dużych grup na jego przeciwległych obrzeżach. Jednakże wystarczy zmienić sposób przedstawienia tego grafu, by zobaczyć, że takie postępowanie będzie po prostu błędne.



Rysunek 6.8: Różne sposoby przedstawienia tego samego grafu.
Źródło: http://www.techques.com/question/1-6558965/Graph-drawing-software

Kolejna metoda wizualizacji o nazwie drzewo stożkowe (*ang. cone tree*) [53] również wykorzystuje przestrzeń trójwymiarową. Jest to dendrogram przypominający swoją strukturą



Rysunek 6.9: Przykład drzewa stożkowego.
Źródło: http://www-graphics.stanford.edu/papers/webviz/figs/euc.gif

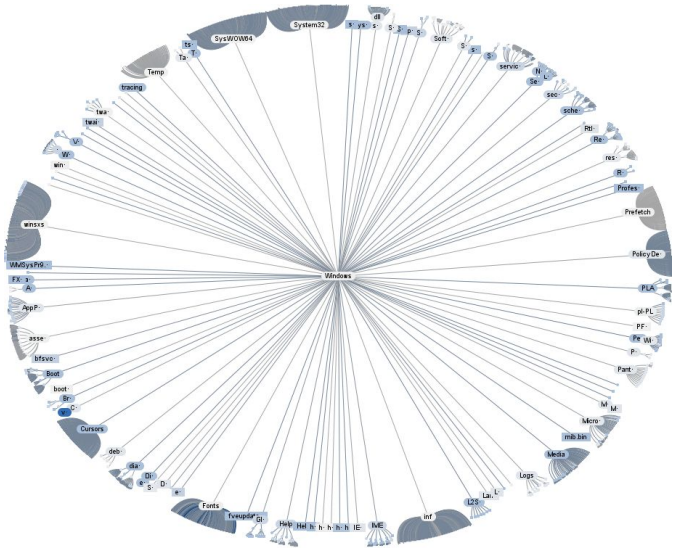
szereg połączonych stożków, które mogą być obracane, aby lepiej przyjrzeć się poszczególnym elementom. Budowa drzewa rozpoczyna się od korzenia (symbolizuje go element, wokół którego wizualizowana jest dalsza hierarchia) w taki sposób, że wszystkie węzły będące jego potomkami rozstawione są w równej odległości względem siebie, formując stożki. Proces ten jest powtarzany dla każdego elementu hierarchii, zmniejszając jednocześnie promień stożka na każdym poziomie hierarchii [39]. Przykładowe drzewo stożkowe zilustrowano na rysunku 6.9. Wadą tej metody jest niewątpliwie trudność w manipulacji takim drzewem, w przypadku gdy posiada ono dużo węzłów (jak podano w [53] technika traci na czytelności, dla struktury posiadającej powyżej 1000 węzłów).

W literaturze przedmiotu [27] można również spotkać szereg technik wykorzystujących przestrzeń dwuwymiarową, służących wizualizacji struktury skupień. Są to różne wariacje techniki map prostokątów oraz metody wykorzystujące wykresy polarne (bądź do nich podobne)[12]. Zalety i wady wykorzystania tych technik, w kontekście wizualizacji złożonych struktur hierarchicznych, zostaną omówione na przykładzie graficznego przedstawienia zajętości miejsca przez pliki i katalogi, znajdujące się wewnątrz katalogu Windows komputera autora.

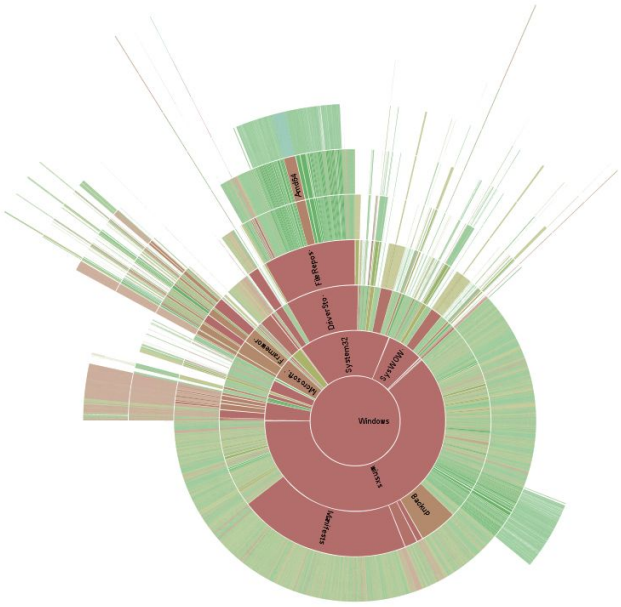
Podejście pierwsze czyli drzewo hiperboliczne (*ang. hiperbolic tree*), przedstawione na rysunku 6.10, wykorzystuje geometrię hiperboliczną, dzięki czemu graf ma więcej miejsca niż w przypadku zastosowania geometrii euklidesowej, ponieważ obwód okręgu na płaszczyźnie hiperbolicznej rośnie wykładniczo wraz ze wzrostem promienia (a nie liniowo). Zatem hierarchie (które mają tendencję do rozszerzenia wykładniczo z głębokością) mogą być rozmieszczone w przestrzeni hiperbolicznej w jednolity sposób tak, że odległość (mierzona w geometrii hiperbolicznej) między rodzicami, dziećmi i rodzeństwem jest w przybliżeniu taka sama na każdym poziomie hierarchii [38]. W środku wizualizacji zawsze pozostaje analizowany w danej chwili węzeł (będący w centrum zainteresowania użytkownika). Diagram ten jednakże nie wykorzy-

[12]Wszystkie omawiane poniżej techniki wizualizacji można zobaczyć i własnoręcznie przetestować, korzystając z oprogramowania Treeviz, dostępnego pod adresem: http://www.randelshofer.ch/treeviz/.

stuje w sposób efektywny, dostępnej przestrzeni roboczej.



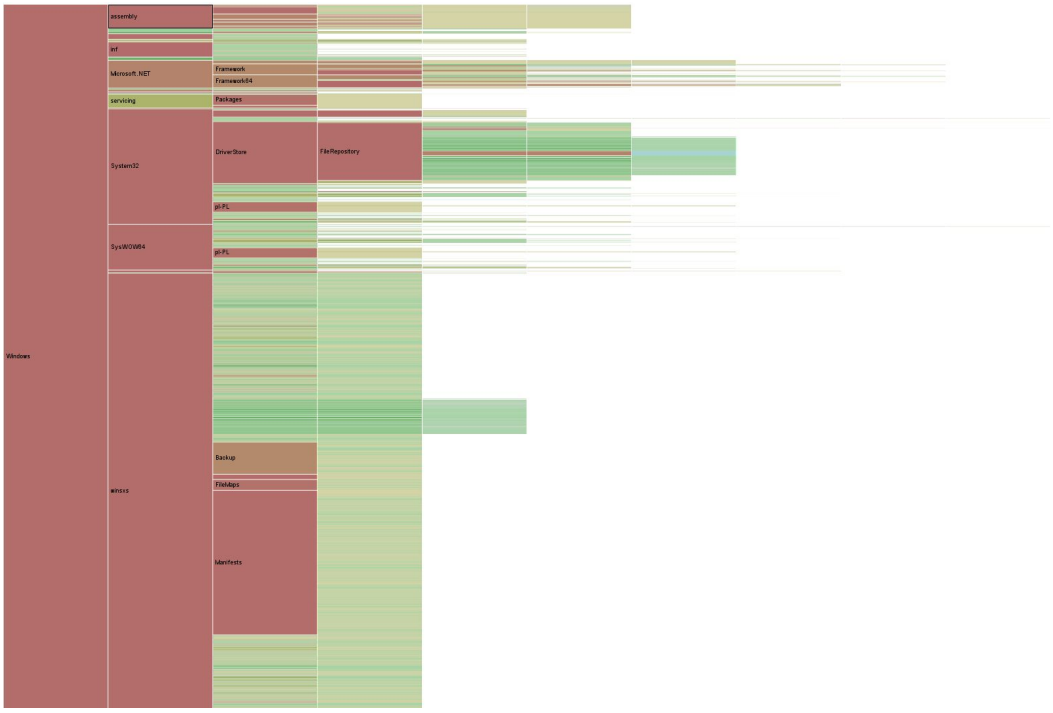
Rysunek 6.10: Przykład drzewa hiperbolicznego.
Źródło: Opracowanie własne



Rysunek 6.11: Przykład techniki sunburst.
Źródło: Opracowanie własne

Sunburst (*ang. sunburst tree*) jest techniką wypełniającą przestrzeń w sposób radialny [62]. Początkowy element całej hierarchii (w tym przypadku katalog Windows) znajduje się w samym centrum obszaru roboczego, co pokazano na rysunku 6.11. Kolejne poziomy hierarchii rysowane są w oddaleniu od centrum. Każdy poziom posiada równą szerokość, ale na podstawie

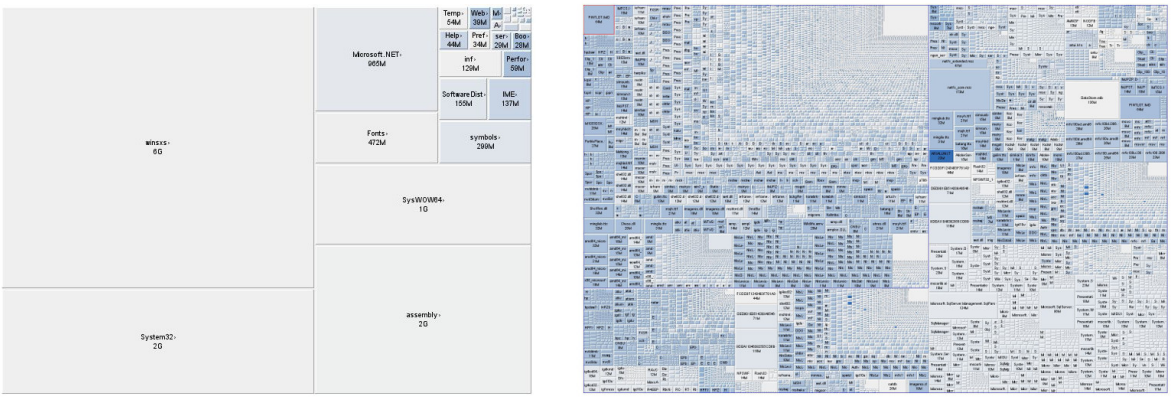
rozpiętości wycinków koła, można oszacować wielkość parametru przez nich reprezentowanego – w tym przypadku zajętość miejsca przez plik bądź katalog. Niestety niewielkie elementy, zajmujące dalsze pozycje w hierarchii (na obrzeżach), są tutaj dość słabo widoczne i mogą zostać pominięte przy analizie. Wariantem metody sunburst jest tzw. diagram typu sunray (*ang. sunray tree*), w którym to liście wizualizowane są jako promienie (słoneczne). Nie wpływa to jednak znacząco na czytelność.



Rysunek 6.12: Przykład diagramu typu icicle.
Źródło: Opracowanie własne

Kolejna technika wizualizacji skupień, spotykana w zasobach literaturowych, nosi nazwę diagramu icicle (*ang. icicle tree*). Jest to transformacja techniki sunburst, która zamiast współrzędnych biegunowych wykorzystuje kartezjańskie. Przykład tej techniki został zaprezentowany na rysunku 6.12. Kolor symbolizuje czas ostatniej modyfikacji, natomiast pole reprezentuje wielkość danego zasobu. Diagram jest lepiej dostosowany do formatów ekranów komputerowych, ale w dalszym ciągu pozostaje dużo niezagospodarowanego miejsca. Podobnie jak poprzednie metody, diagram ten ma swoją pochodną o nazwie iceray (*ang. iceray tree*) i przedstawia liście w formie promieni. Jest to liniowa wersja diagramu sunray, która niestety zachowuje wady swojego pierwowzoru.

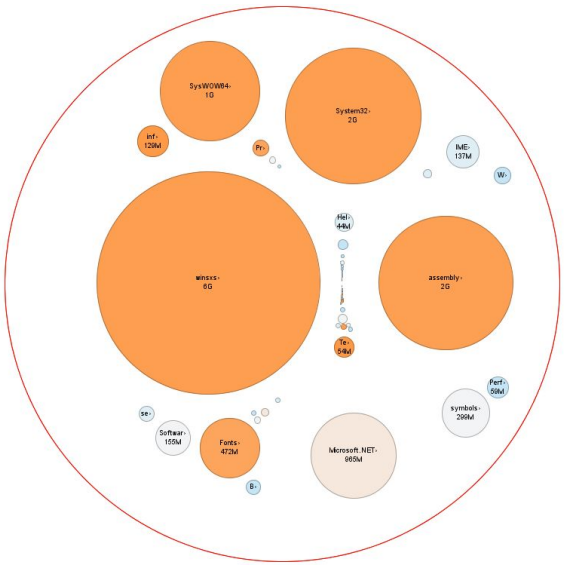
Mapa prostokątów (ang. *rectangular treemap*) jest techniką wizualizacji struktur hierarchicznych, dzielącą rekurencyjnie dostępną przestrzeń na szereg prostokątów, których wielkość jest zależna od wartości analizowanej danej – w naszym przypadku zajętości zasobu dyskowego. W przykładzie zobrazowanym na rysunku 6.13a, wizualizowany jest jedynie najwyższy poziom hierarchii, ponieważ dla dużych zbiorów jest to zwykle bardziej czytelne podejście. Niemniej sama technika może przedstawiać jednocześnie więcej poziomów hierarchii naraz, co



a) jeden poziom hierarchii

b) wszystkie poziomy hierarchii

Rysunek 6.13: Przykład klasycznej mapy prostokątów.
Źródło: Opracowanie własne



Rysunek 6.14: Przykład kolistej mapy prostokątów.
Źródło: Opracowanie własne

pokazuje rysunek 6.13b. Dodatkowo, ze względu na fakt, że opisywana metoda wykorzystuje cały dostępny obszar roboczy ekranu, pozwala ona przedstawić graficznie nawet bardzo skomplikowaną strukturę skupień, dzięki czemu została wybrana jako narzędzie wykorzystane przy analizie rzeczywistych danych złożonych, będących przedmiotem badań.

Kolista mapa prostokątów (ang. *circural treemap*), przedstawiona na rysunku 6.14, jest przykładem diagramu wykorzystującego zawieranie, w przeciwieństwie do przylegania elementów, celem wizualizacji hierarchii. Ponownie jak wcześniej pola, poszczególnych kół utożsamiają rozmiar zasobów na dysku. Diagram ten nie wykorzystuje zbyt efektywnie dostępnego obszaru roboczego, aczkolwiek dzięki temu można z łatwością porównać wielkość poszczególnych elementów. Ponadto jeżeli reprezentowane są wszystkie poziomy hierarchii, przybliżanie (ang.

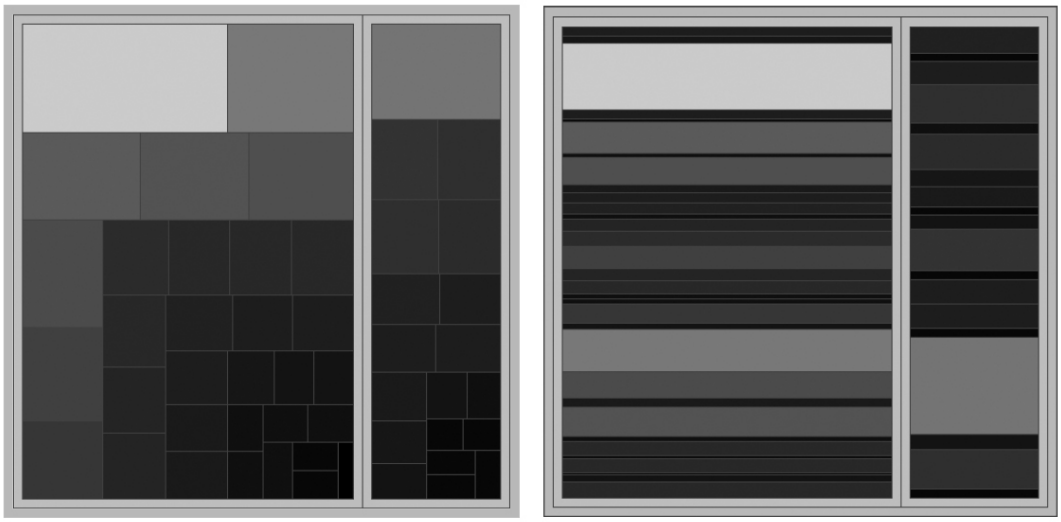
zooming) dowolnego fragmentu wizualizacji może odbywać się relatywnie szybko, ponieważ nie ma potrzeby powtórnego przeliczenia otrzymanej struktury (co może mieć miejsce, jeżeli wykorzystywana jest klasyczna mapa prostokątów). Niestety przy dużej liczbie elementów i poziomów hierarchii, może dojść do sytuacji, w której koła podobnych rozmiarów reprezentują zupełnie różne zasoby np. pliki różniące się zajętością pamięci.

6.4 Generowanie map prostokątów

Dostępny obszar roboczy można podzielić na prostokąty na wiele sposobów — dla ustalonej liczby prostokątów można zaproponować wiele różnych ułożeń, pozostawiając bez zmian ich pole powierzchni. Dlatego też algorytmy generowania takiego podziału, ocenia się na podstawie trzech parametrów: utrzymania proporcji (ang. *aspect ratios*) między długością a szerokością prostokątów, zachowania określonej kolejności ich tworzenia oraz stabilności rozumianej najczęściej jako liczba zmian dokonywanych w wizualizacji przy aktualizacji źródłowego zbioru danych. Pierwsza miara oceny czyli tzw. średni współczynnik proporcji (ang. *average aspect ratio*) układu map prostokątów – to średnia arytmetyczna stosunku długości i szerokości wszystkich prostokątów (na najniższym poziomie hierarchii, jeżeli wizualizowane są wszystkie). W źródłach literaturowych [11, 15] podaje się optymalną wartość tego współczynnika jako niską i bliską jedynce. Dzięki temu utworzone prostokąty są bardziej zwarte, a porównanie powstałych kształtów jest łatwiejsze dla analityka posługującego się tą metodą wizualizacji. Zostanie to zobrazowane na następującym przykładzie. Rysunek 6.15 przedstawia różnicę między dwoma popularnymi sposobami tworzenia podziału: *Slice and Dice* [58] oraz *Squarified* [11]. Zadaniem obu algorytmów jest podział identycznego obszaru roboczego na dokładnie taką samą liczbę prostokątów. Dodatkowo dla poprawy czytelności zastosowano skalę szarości – jaśniejszy odcień odznacza większe pole powierzchni danego obszaru. W przypadku pierwszej techniki tworzone są często bardzo cienkie prostokąty, co negatywnie wpływa na czytelność wizualizacji i może być przyczyną sformułowania błędnych wniosków na temat prezentowanych danych. Jest to zatem najważniejszy parametr oceny jakości wizualizacji, w kontekście jej zastosowania do rzeczywistych zbiorów danych złożonych.

Zachowanie relacji między kolejnością prostokątów a odpowiadającym im obiektom ze zbioru danych, może być przydatne w sytuacji, gdy istnieje z góry narzucony ranking tych obiektów np. prezentowane są dane finansowe spółek notowanych na giełdach i każda z nich posiada określoną rangę. W przypadku danych telekomunikacyjnych będących przedmiotem badań, trudno jest ustalić podobną zależność, dlatego ten parametr oceny nie jest uznawany za istotny. Podobnie stabilność rozpatrywana jest jako kluczowa, w sytuacji gdy zbiór danych podlega częstym aktualizacjom. Wówczas niska stabilność, charakteryzująca się znacznymi zmianami układu map prostokątów przy niewielkich modyfikacjach danych źródłowych, znacząco utrudnia wychwycenie przez analityka dokonanych aktualizacji i ich ewentualnego wpływu na dalszą analizę. Autor pracy dysponuje jednak danymi rzeczywistymi, które pochodzą ze ściśle określonego przedziału czasowego, w związku z czym nie ma możliwości, aby były one zmieniane i porównywane na bieżąco.

Pomysłodawcą idei map prostokątów był Ben Shneirderman, który w 1991 roku zaprezen-



Rysunek 6.15: Porównanie algorytmów *Slice and Dice* oraz *Squarified* pod kątem średniego współczynnika proporcji całego układu.

Źródło: [15]

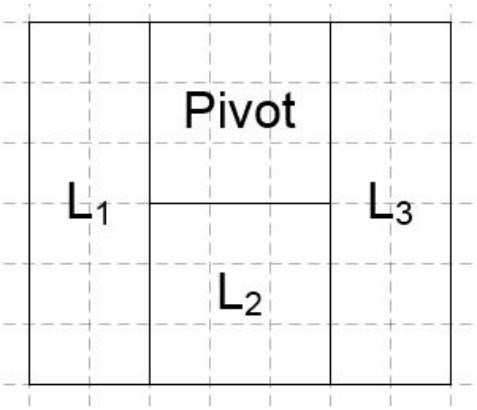
tował algorytm *Slice and Dice*. Postawił sobie za cel stworzenie dwuwymiarowej techniki wizualizacji struktur hierarchicznych, która charakteryzuje się następującymi cechami: efektywnie wykorzystuje dostępną przestrzeń roboczą, posiada możliwość interakcji z użytkownikiem, jest łatwa w zrozumieniu i estetycznie wygląda [30]. W proponowanym podejściu, dla określonego poziomu wizualizowanego drzewa, dostępny obszar dzielony jest na prostokąty (których pole powierzchni tożsame jest z wagą lub określonym parametrem ilościowym) w jednym, ustalonym uprzednio kierunku np. poziomo. Następnie prostokąty, symbolizujące elementy kolejnego poziomu hierarchii, układane są w przeciwny sposób czyli pionowo. Proces jest rekurencyjnie powtarzany, aż do osiągnięcia ostatniego poziomu. Niestety (jak pokazano na rysunku 6.15b) takie postępowanie sprawdza się najczęściej tylko dla mniejszej liczby elementów o podobnych rozmiarach – w przeciwnym przypadku tworzone są cienkie, trudne w porównaniu prostokątne pasy. Sam algorytm charakteryzuje się bowiem bardzo wysokim średnim, współczynnikiem proporcji, co zweryfikowano eksperymentalnie w [15, 11]. Wyklucza to jego zastosowanie do dużych zbiorów danych złożonych. Do jego zalet należy jednak zaliczyć zachowanie kolejności wizualizacji elementów (zgodnie z danymi źródłowymi) jak również wysoką stabilność.

W literaturze przedmiotu [59, 67] spotyka się kilka modyfikacji przedstawionej techniki. Bruls, Huizing oraz van Wijk w 2000 stworzyli algorytm ułożenia o nazwie *Squarified* [11], który za pomocą prostej heurystyki, próbuje generować prostokąty o współczynniku proporcji bliskim jedynce. Zatem każdy prostokąt $r \in R$ (gdzie R to zbiór wszystkich prostokątów jakie powinny zostać utworzone) powinien spełniać następujące kryterium:

$$\forall_{r \in R} \frac{\max(szer_r,\ dlg_r)}{\min(szer_r,\ dlg_r)} \approx 1, \tag{6.1}$$

gdzie $szer_r$ to szerokość prostokąta, a dlg_r to jego długość. Oczywiście znalezienie optymalnego rozwiązania (dla takiego ułożenia całej hierarchii) to problem NP-trudny [15], dlatego też

stosuje się pewne przybliżenie. Mianowicie zamiast analizować wszystkie poziomy hierarchii naraz, algorytm generuje prostokąty wyłącznie dla elementów jednego poziomu (rozpoczynając od najwyższego). Daje to dobre warunki początkowe do dalszego podziału utworzonych prostokątów, tym razem dla niższego poziomu hierarchii. Algorytm produkuje bardzo dobre ułożenie w kontekście średniego współczynnika proporcji, jednakże nie zachowuje on uporządkowania (kolejności) elementów. Dodatkowo stabilność jest na niskim poziomie – aktualizacja zbioru danych może całkowicie przestawić prostokąty na wizualizacji, utrudniając użytkownikom orientację.



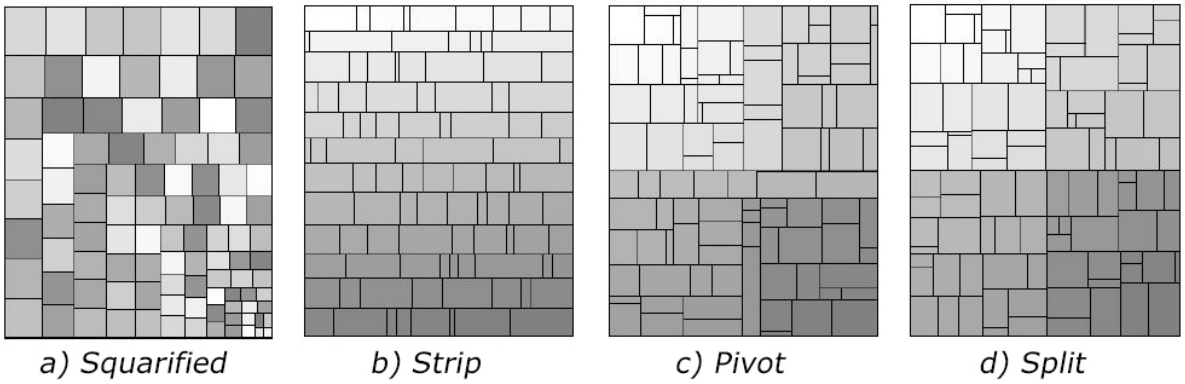
Rysunek 6.16: Schemat postępowania przy algorytmach typu *Ordered Treemap*.
Źródło: Opracowanie własne

Shneiderman i Wattenberg [59] zaproponowali zmodyfikowany algorytm *Ordered Treemap*, który minimalizuje proporcje między długością a szerokością prostokątów, jednocześnie zachowując pewne uporządkowanie danych na każdym poziomie hierarchii z osobna. Zasadę jego działania można porównać do powszechnie znanego algorytmu sortowania QuickSort. Niech obiekty zbioru wejściowego mają przypisane indeksy (numery), symulując strukturę uporządkowanej listy. Wspomniana lista jest dzielona na cztery części: trzy mniejsze listy (L_1, L_2, L_3) i jeden specjalny element zwany dzielącym (ang. *pivot*). *Pivot* wizualizowany jest jako kwadrat, natomiast zawartość pozostałych list reprezentowana jest rekursywnie we właściwych, zaznaczonych na schemacie 6.16 prostokątach. L_1 składa się ze wszystkich elementów o indeksach mniejszych od indeksu elementu środkowego. Podobnie L_2 zawiera elementy o numerach mniejszych niż te w L_3, ale średni współczynnik proporcji prostokąta (reprezentującego wspomniane elementy) jest jak najbardziej zbliżony do jedynki. Takie postępowanie powtarzane jest rekurencyjnie dla list L_1, L_2, L_3, uzyskując ostatecznie uporządkowany podział. Problematyczny może być dobór elementu środkowego. Autorzy algorytmu [59] wyszczególnili dwa sposoby doboru *pivota*: jako największy (pod względem reprezentowanego parametru ilościowego) obiekt na liście lub element środkowy listy. Pierwszy sposób charakteryzuje się nieco lepszym średnim współczynnikiem proporcji, natomiast drugi posiada lepszą stabilność.

Kolejny algorytm o nazwie *Strip* jest pewnym uproszczeniem techniki *Squarified*, która zachowuje kolejność wizualizowanych elementów [59]. Wyróżnia się on w dwóch aspektach. Technika *Squarified* przy dodawaniu każdego prostokąta do wizualizacji, bada czy nowy element lepiej jest rozłożyć pionowo bądź poziomo (uwzględniając kryterium 6.1), natomiast *Strip* nie

zmienia kierunku ułożenia. Ponadto *Squarified*, w celu zachowania jak najlepszego średniego współczynnika proporcji, sortuje wizualizowane prostokąty od najmniejszego do największego (biorąc pod uwagę ich pola powierzchni). Krok ten nie występuje w algorytmie *Strip* [15].

Ostatnim przedstawicielem algorytmów zachowujących kolejność elementów jest *Split*. Stara się on poprawić wadę swoich poprzedników (czyli relatywnie wysokie wartości średniego współczynnika proporcji), zachowując ich zalety. Jego ogólna zasada działania jest następująca. Przy założeniu, że istnieje uporządkowana lista elementów L (które mają być wizualizowane), jest ona dzielona na dwie listy L_1 i L_2 w taki sposób, że suma wartości analizowanego parametru ilościowego (co jest reprezentowane przez pola prostokątów) jest w przybliżeniu taka sama na obu listach. Dostępny obszar roboczy wizualizacji jest następnie dzielony na dwa prostokąty (pionowo bądź poziomo, w zależności od tego czy jest on szerszy czy dłuższy), które stanowią odzwierciedlenie list L_1 oraz L_2. Przedstawiony proces powtarza się rekurencyjnie dla L_1 oraz L_2.

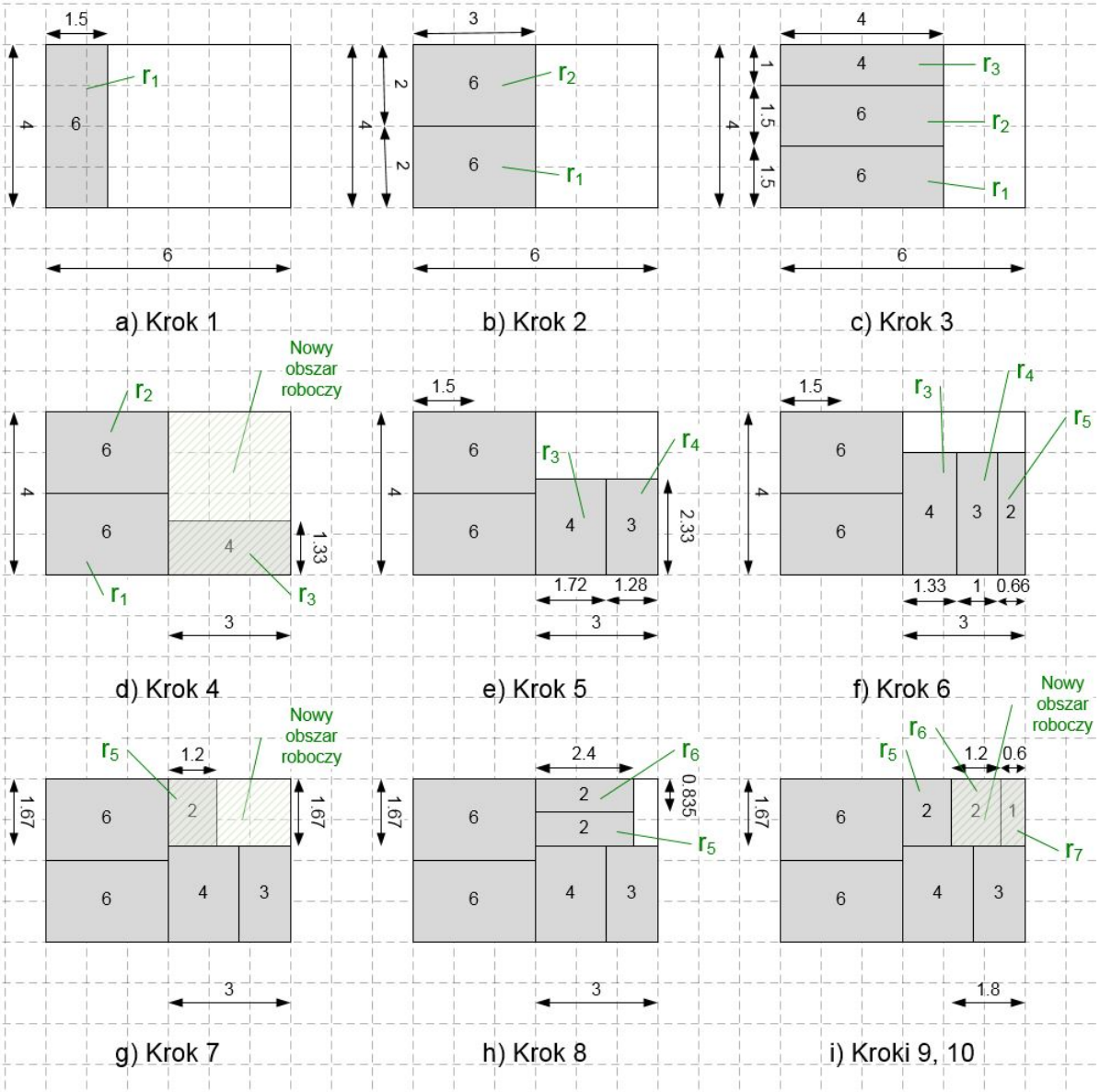


Rysunek 6.17: Porównanie najpopularniejszych technik generowania map prostokątów. Źródło: [15]

Rysunek 6.17 przedstawia poglądowe porównanie rezultatów czterech omawianych technik generowania map prostokątów. Skala szarości została zastosowana, by odwzorować kolejność elementów w źródłowym zbiorze danych. Ciemniejszy odcień na wizualizacji oznacza dalszą pozycję obiektu w zdefiniowanym uporządkowaniu. Najlepsze wyniki pod kątem średniego współczynnika proporcji (a tym samym łatwości porównywania prostokątów ze sobą) prezentuje technika *Squarified*. Wśród metod uwzględniających kolejność elementów, technika *Split* posiada najlepszy średni współczynnik proporcji. Algorytm rozmieszczenia *Strip* generuje miejscami dość wąskie prostokąty, co utrudnia ich lokalizację czy porównanie. Przedstawione wnioski zostały również potwierdzone eksperymentalnie w [15], gdzie znajdują się szczegółowe informacje dotyczące wspomnianych algorytmów i przeprowadzonych badań.

Balzer i Deussen [5] zaproponowali wykorzystanie idei diagramów Woronoja do reprezentowania danych, zamiast prostokątów. Uzyskuje się zatem podział przestrzeni na (zazwyczaj wypukłe) wielokąty, przez co bazowy obszar roboczy może przyjąć dowolny kształt – trójkątny, owalny itp. Jednak porównywanie elementów na tego typu diagramie jest utrudnione, ze względu na ich różne kształty – dwie komórki Woronoja mogą mieć taką samą powierzchnię, natomiast różnić się znacząco wyglądem. Ponadto jak stwierdzono w [50], brak jest teoretycz-

nych gwarancji odnośnie wartości średniego współczynnika proporcji. Jeżeli obszar roboczy jest okrągły, często wykorzystywane są wymienione wcześniej koliste mapy prostokątów (ang. *circural treemaps*), które choć wizualnie wydają się być estetyczne i dobrze prezentują zawieranie się poziomów hierarchii, to jednak dużo przestrzeni jest marnowanej przy procesie podziału kół na mniejsze.



Rysunek 6.18: Przykład działania techniki *Squarified*.
Źródło: Opracowanie własne

Warto także wspomnieć o rozszerzeniu koncepcji map prostokątów do przestrzeni trójwymiarowej, poprzez budowę tzw. kostki informacyjnej (ang. *information cube*) Rekimoto i Green'a [52] lub wykorzystując oprogramowanie *StepTree* stworzone przez Bladh'a, Carr'a oraz Scholl'a [10]. W pierwszym przypadku wykorzystywana jest przeźroczystość, dzięki czemu

możliwe jest stworzenie trójwymiarowej kostki, która zawiera w sobie inne, reprezentując tym samym całą hierarchię. Druga koncepcja jest nieco bliższa oryginałowi – prostokąty należące do różnych poziomów hierarchii, układane są jeden na drugim (symulując stos) w przestrzeni trójwymiarowej. Dzięki obracaniu, przybliżaniu oraz nadawaniu różnego stopnia przeźroczystości wizualizowanym elementom, reprezentacja ta jest nieco prostsza w zrozumieniu i analizie od poprzedniej. Mimo wszystko, przy dużej liczbie prezentowanych elementów, metody te tracą na czytelności, przez co również nie będą miały zastosowania w niniejszej pracy.

Biorąc pod uwagę wady i zalety przedstawionych podejść do generowania różnych układów prostokątów, zdecydowano się wybrać technikę *Squarified* jako najlepszy algorytm do reprezentowania struktury skupień (która będzie wygenerowana przez zastosowanie gęstościowej metody analizy skupień). Technika ta posiada najlepszy średni współczynnik proporcji [15], co jest bardzo istotne podczas porównywania skupień i wyborze tych, na których powinna się skupić dalsza analiza. Działanie *Squarified* zostanie przedstawione na rysunku 6.18.

Niech dostępny jest obszar roboczy o wymiarach 6×4 jednostki oraz dostępnych jest siedem prostokątów (reprezentujących obiekty danych) o polach powierzchni odpowiednio $6, 6, 4, 3, 2, 2, 1$. Zbiór elementów jest już posortowany malejąco (w przeciwnym przypadku należałoby go uporządkować). Dzięki temu uzyskuje się lepsze (pod kątem utrzymania proporcji) ułożenia [11]. Pierwszym krokiem algorytmu *Squarified* jest wybór początkowego sposobu rozkładania elementów – pionowo bądź poziomo – w zależności od wielkości obszaru roboczego. Ponieważ obszar, który należy wypełnić jest szerszy niż dłuższy, pierwszy prostokąt r_1 umieszczany jest pionowo. Należy zatem wyznaczyć, jakie będą wymiary nowego prostokąta. Wiadomo, że obszar roboczy posiada długość czterech jednostek, dlatego szerokość nowego prostokąta (o polu powierzchni równym sześć) można wyliczyć jako:

$$\begin{aligned} szer_{r1} \cdot 4 &= 6, \\ szer_{r1} &= \frac{6}{4} = 1,5 \end{aligned}$$

Nowy prostokąt będzie posiadał zatem wymiary $1,5 \times 4$ (co zaznaczono na rysunku 6.18a). Następnie należy wyliczyć, jaki jest średni współczynnik proporcji utworzonego prostokąta, który dla opisywanej techniki powinien być bliski jedynce. Posługując się zależnością z kryterium 6.1) otrzymujemy:

$$proporcje_{r1} = \frac{\max(1,5,\ 4)}{\min(1,5,\ 4)} = \frac{4}{1,5} \approx 2,666 \tag{6.2}$$

W drugim kroku algorytmu ponownie należałoby zweryfikować kierunek układania prostokątów (pionowo bądź poziomo), tak by optymalizować kryterium 6.1. Jednakże został ustawiony dopiero jeden prostokąt, dlatego nie ma jak porównać aktualnego współczynnika proporcji z poprzednim. Będzie to wykonywane dopiero od kroku trzeciego. Algorytm kontynuuje zatem nakładanie prostokątów pionowo. Po nałożeniu kolejnego prostokąta r_2 na aktualny r_1, uzyskuje się zajętą powierzchnię o wielkości 12 jednostek, ponieważ każdy z prostokątów ma pole równe 6 jednostkom (zgodnie z rysunkiem 6.18b). Dlatego należy (podobnie jak poprzednio) wyznaczyć prawidłową szerokość oraz długość dla obu prostokątów (by zachować pola

powierzchni bez zmian) jako:

$$szer_{r1r2} \cdot 4 = 12,$$
$$szer_{r1r2} = \frac{12}{4} = 3,$$
$$dlg_{r1} = \frac{6}{3} = 2,$$
$$dlg_{r2} = \frac{6}{3} = 2,$$

gdzie $szer_{r1r2}$ oznacza całkowitą szerokość powierzchni zajmowaną przez prostokąty $r1$ i $r2$, natomiast dlg_{r1}, dlg_{r2} odpowiednio nową długość prostokątów $r1$ oraz $r2$. Prostokąty zmieniają więc swoje wymiary na 3×2 jednostki. Wyznaczane są ponownie proporcje dla nowo-dodanego prostokąta:

$$proporcje_{r2} = \frac{\max(3,\ 2)}{\min(3,\ 2)} = \frac{3}{2} = 1,5 \tag{6.3}$$

Średni współczynnik proporcji dla całej wizualizacji, po dołożeniu prostokąta $r2$, uległ poprawie (jest bliższy jedynce) względem wyliczeń z równania 6.2, dlatego też algorytm kontynuuje umieszczanie prostokątów pionowo.

W kroku trzecim dokładany jest prostokąt r_3 o polu powierzchni równym cztery, zatem obszar zajmowany przez wszystkie prostokąty wyniesie 16 jednostek. Następuje zmiana szerokości i długości prostokątów:

$$szer_{r1r2r3} \cdot 4 = 16,$$
$$szer_{r1r2r3} = \frac{16}{4} = 4$$
$$dlg_{r1} = \frac{6}{4} = 1,5$$
$$dlg_{r2} = \frac{6}{4} = 1,5$$
$$dlg_{r3} = \frac{4}{4} = 1$$

Prostokąty $r1$ i $r2$ zmieniają swoje wymiary na $4 \times 1,5$, natomiast $r3$ ze względu na nieco niższe pole powierzchni ma wymiary 4×1 jednostki (zgodnie z rysunkiem 6.18c). Ponownie badana jest zmiana dla średniego współczynnika proporcji, jaką wnosi umieszczenie prostokąta $r3$:

$$proporcje_{r3} = \frac{\max(4,\ 1)}{\min(4,\ 1)} = \frac{4}{1} = 4 \tag{6.4}$$

Widoczne jest wyraźne pogorszenie sytuacji – średni współczynnik proporcji wzrósł do czterech względem poprzedniego, równego półtorej. Dlatego też takie ułożenie nie jest akceptowane i algorytm cofa się do ustawienia prostokątów z kroku drugiego, a następnie je blokuje. Oznacza to, że żadne zmiany nie będą już wprowadzane do obszaru wizualizacji zajmowanego przez prostokąty $r1$ i $r2$. Jako nowy obszar roboczy do umieszczania prostokątów, traktowana jest

pozostała wolna część, czyli przestrzeń o wymiarach 3×4 (zobacz rysunek 6.18d). Ponieważ zmieniła się wielkość obszaru roboczego, należy ustalić kierunek układania nowych prostokątów – tym razem będą one układane poziomo względem siebie, ponieważ dostępna przestrzeń jest dłuższa niż szersza.

W kroku czwartym, wprowadzany jest prostokąt $r3$ do nowego obszaru roboczego (jak pokazano na rysunku 6.18d). Wyznaczane są tym samym jego nowe wymiary:

$$dlg_{r3} \cdot 3 = 4,$$

$$dlg_{r3} = \frac{4}{3} = 1,333$$

oraz wpływ na średni współczynnik proporcji:

$$proporcje_{r3} = \frac{\max(3,\ 1,33)}{\min(3,\ 1,33)} = \frac{3}{1,33} \approx 2,251 \tag{6.5}$$

Krok piąty jest analogiczny do poprzedniego – wprowadzany jest prostokąt $r5$, umieszczony poziomo obok $r4$ (zgodnie z rysunkiem 6.18e). Aktualizowane są wymiary obu prostokątów i wyznaczany jest współczynnik proporcji, który wynosi w przybliżeniu $1,820$, widać więc poprawę względem poprzedniego $(2,251)$, dlatego kolejne prostokąty dalej będą umieszczane poziomo. W kroku szóstym (który ilustruje rysunek 6.18f), po dołożeniu prostokąta $r5$, średni współczynnik proporcji ulega pogorszeniu i wynosi około $4,545$. Dlatego też algorytm cofa ułożenie, do tego z kroku piątego i jest ono blokowane. Zatem żadne zmiany w położeniu prostokątów $r3$ oraz $r4$ nie są już możliwe. Wybierany jest również nowy obszar roboczy o wymiarach $3 \times 1,67$ jednostek (zgodnie z rysunkiem 6.18g), który jest szerszy niż dłuższy, dlatego algorytm rozpoczyna układanie prostokątów pionowo.

W kroku siódmym, dokładany jest prostokąt $r5$ do nowego obszaru roboczego i wyznaczany jest dla niego współczynnik proporcji równy $1,391$. Krok ósmy (pokazany na rysunku 6.18h) wygląda analogicznie. Niestety takie ułożenie pogarsza średni współczynnik proporcji wizualizacji do $2,874$ (względem poprzedniego $1,391$), więc algorytm cofa ułożenie do wykonanego w kroku siódmym i blokuje prostokąt $r5$. Pozostałe prostokąty rozkładane są poziomo na obszarze o wymiarach $1,8 \times 1,67$, aż do wypełnienia całej przestrzeni. Ostatecznie, średni współczynnik proporcji całej wizualizacji wynosi około $2,783$. Należy jednak pamiętać, że w przytoczonym przykładzie, dla takiego zbioru prostokątów i obszaru roboczego, nie jest możliwe uzyskanie idealnej wartości tego współczynnika równej jeden. Jak zostało pokazane w [11], zastosowanie algorytmu *Slice and Dice* dla identycznego zbioru, daje wizualizację o średnim współczynniku proporcji równym 16 bądź 31, w zależności od wyboru podziału pionowego lub poziomego.

Dla omówionego przykładu, suma pól wizualizowanych prostokątów była tożsama z polem powierzchni dostępnego obszaru roboczego, w skutek czego wypełniły one całą dostępną przestrzeń. Niestety taka sytuacja nie zawsze ma miejsce dla dużej liczby elementów i wówczas wprowadzane są często odstępy między prostokątami oraz są one nieco zniekształcane przez skalowanie, tak by wypełnić cały obszar roboczy. Jest to jednak sytuacja niepożądana, gdy wizualizacja jest wykorzystywana do reprezentowania skupień rzeczywistych danych, ponieważ

porównując wielkości dwóch prostokątów można dojść do mylnych wniosków. Dlatego też autor niniejszej pracy postanowił zaimplementować algorytm wizualizacji tak, by nie naruszał proporcji prostokątów, nawet kosztem pozostawienia pustego miejsca na obszarze roboczym.

6.5 Podsumowanie

Niniejszy rozdział skupia się na analizie metod reprezentacji skupień danych złożonych. Opisuje on wady i zalety klasycznych technik reprezentacji skupień (w postaci dendrogramu czy diagramu Woronoja), jak również rzadziej stosowanych metod (jak diagram typu sunray). Rozważania poparto prostymi przykładami zastosowania przedstawionych metod wizualizacji – wybór różnych wykorzystywanych zbiorów danych został podyktowany ograniczeniami i specyfiką stosowanych narzędzi, które nie umożliwiały bezpośredniego wczytania danych rzeczywistych (np. ze względu na różnice w wykorzystywanych strukturach).

Szczególnym aspektem analiz było również omówienie koncepcji generowania map prostokątów. Opisywana technika znalazła zastosowanie w autorskim systemie *DensGroup* do wydobywania wiedzy z danych złożonych, jako podstawowy sposób wizualizacji skupień. Ze względu na fakt, iż algorytm *Squarified* generuje najbardziej zwarte i relatywnie proste w analizie porównawczej prostokąty, zdecydowano się na ostateczny wybór tego wariantu zamiast innych opisywanych.

Dodatkowo omówiono proces graficznej analizy eksploracyjnej, wraz z uzasadnieniem przydatności oraz wpływu wiedzy dziedzinowej i zdolności kognitywnych człowieka na ostateczne rezultaty procesu wydobywania wiedzy z danych. Zauważalny trend wykorzystywania technik wizualizacyjnych w ekstrakcji wiedzy, zdaje się zaprzeczać tezie, że jest to w dużej mierze autonomiczny i pozbawiony nadzoru proces.

Rozdział 7

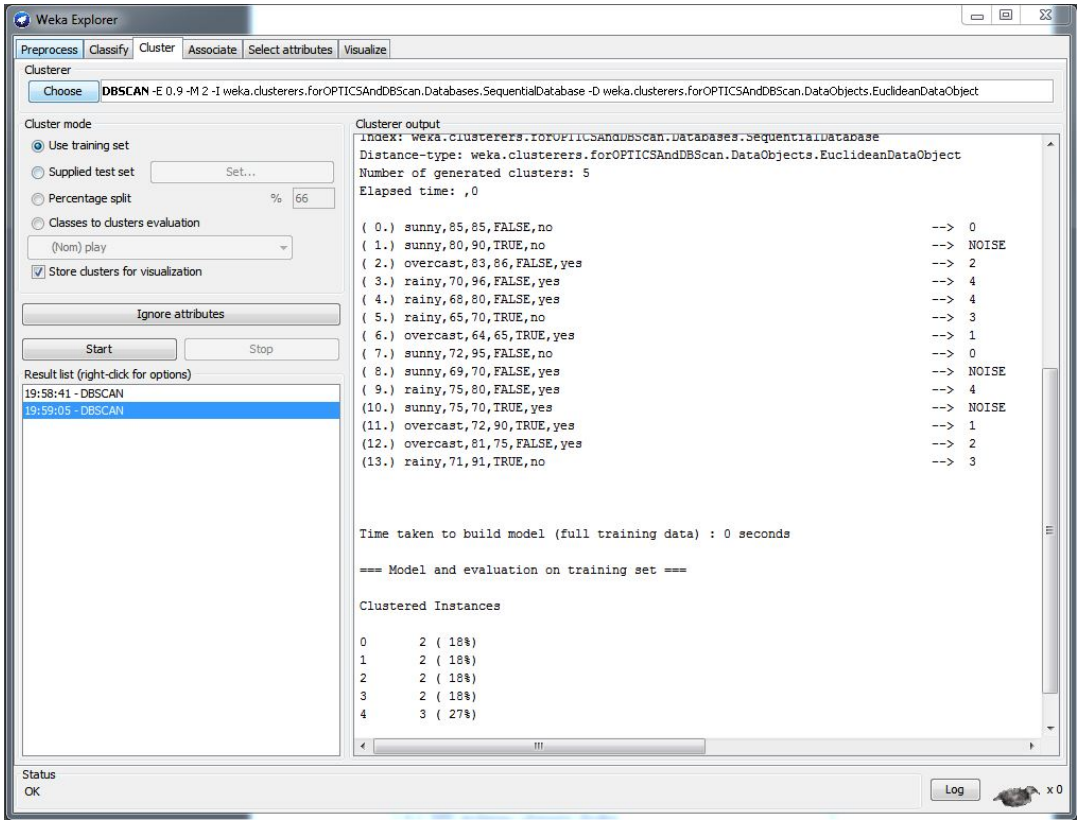
Projekt i implementacja systemu DensGroup

Wydobywanie wiedzy z danych złożonych jest procesem wieloetapowym, który często nie jest możliwy do przeprowadzenia bez udziału wyspecjalizowanego oprogramowania, dedykowanego do analizy danych. Jednakże przeprowadzony w rozdziale 3 przegląd dostępnych rozwiązań programowych wykazał, że mimo dużej liczby oferowanych narzędzi (zarówno komercyjnych jak i bezpłatnych), brak jest systemu udostępniającego bezpośrednio interaktywną metodę graficznej reprezentacji skupień, dostosowaną do wizualizacji dużej liczby grup. Ponadto tylko niewielka liczba programów implementuje bardziej zaawansowane algorytmy analizy skupień (np. gęstościowe) i umożliwia ich zastosowanie do danych opisanych atrybutami ilościowymi oraz jakościowymi. Przykładowo popularny program Weka[1] [88], chociaż posiada zaimplementowane algorytmy *DBSCAN* oraz *OPTICS*, nie ułatwia znacząco wykrycia zależności korzystając z wymienionych metod analizy skupień, przez brak wygodnego interfejsu do przetwarzania i interpretacji wyników tego procesu. Wygląd głównego okna programu, po przeprowadzeniu grupowania gęstościowego (na zbiorze danych pogodowych *weather*[2]), przedstawia rysunek 7.1. Jedynymi istotnymi informacjami, podawanymi przez opisywane oprogramowanie są: przydział do grup poszczególnych obiektów oraz liczba elementów w skupieniach.

Niestety obiekty nie są nawet posortowane zgodnie z ich przynależnością do danego skupienia, jak również nie został zaproponowany jakikolwiek reprezentant grupy. Użytkownik otrzymuje zatem jedynie pewne przyporządkowanie (obiektów do skupień), które wprost nie dostarcza mu nowej wiedzy, ani nie sugeruje występowania (bądź nie) jakichkolwiek zależności czy wzorców w danych. Sytuacja znacząco komplikuje się w przypadku, gdy analizowane są zbiory wielowymiarowe o dużej liczbie obiektów. Wówczas wyniki prezentowane przez Wekę stają się nieczytelne i niemożliwe w interpretacji, bez ich skopiowania i przetworzenia w innym programie. Ponadto, podczas wykonywania eksperymentów na potrzeby niniejszej rozprawy

[1] Program został szczegółowo omówiony w rozdziale 3.

[2] Wspomniany zbiór instalowany jest wraz z oprogramowaniem Weka [88]. Wybrano go do prezentacji przykładów ze względu na prostotę przekazu.



Rysunek 7.1: Przykład działania algorytmów grupowania w programie Weka.
Źródło: Opracowanie własne

stwierdzono, że wspomniane narzędzie Weka w pewnych przypadkach potrafi się zawiesić po kilkunastu godzinach pracy, nie zwracając żadnego wyniku. Są to główne powody, które stanowiły motywację dla autora rozprawy do stworzenia systemu wydobywania wiedzy z danych, z wykorzystaniem technik analizy skupień nazwanego *DensGroup*.

Niniejszy rozdział stanowi uproszczoną dokumentację przestawiającą instalację, wymogi sprzętowe oraz możliwości autorskiego systemu *DensGroup*. Umożliwia to wstępne zapoznanie się z interfejsem wspomnianego narzędzia oraz sposobem jego obsługi.

7.1 Instalacja i wymagania sprzętowe aplikacji DensGroup

Program dedykowany jest dla środowiska wyposażonego w system Microsoft *Windows XP Professional Service Pack 3* lub *Windows 7 Professional Service Pack 1* w wersji 32 lub 64-bitowej[3]. Minimalne wymagania sprzętowe dla systemu DensGroup przedstawiono w tabeli 7.1.

Przy przetwarzaniu dużych zbiorów danych, zalecane jest wykorzystanie komputera wyposażonego w co najmniej 8 GB pamięci RAM oraz 64-bitowy system operacyjny, który umożli-

[3]Możliwe jest stworzenie wersji programu działającej pod innymi systemami operacyjnymi (jak *Linux* czy *Mac OS*), poprzez odpowiednie skompilowanie źródła programu umieszczonego na płycie CD, dołączonej do pracy.

Tabela 7.1: Minimalne wymagania sprzętowe systemu *DensGroup*

Procesor jednordzeniowy o częstotliwości taktowania 1,8 GHz
3 GB pamięci RAM
300 MB wolnego miejsca na dysku twardym
Monitor o rozdzielczości 1366×768 px
Napęd CD-ROM
Klawiatura, mysz

wia dostęp do całej puli adresowej. Program, ze względu na wykorzystanie platformy programistycznej *Qt* [87] oraz tzw. linkowania statycznego, nie wymaga żadnych dodatkowych bibliotek. Wyjątkiem jest jedynie funkcjonalność podłączenia do bazy danych[4], która naturalnie wymaga prawidłowo skonfigurowanego serwera bazodanowego oraz dedykowanego dla niego sterownika ODBC. Program był testowany z serwerem *Microsoft SQL Server 2012 Enterprise*, jednakże istnieje możliwość pracy z produktami innych firm pod warunkiem, że w systemie istnieje odpowiedni sterownik ODBC umożliwiający połączenie z serwerem.

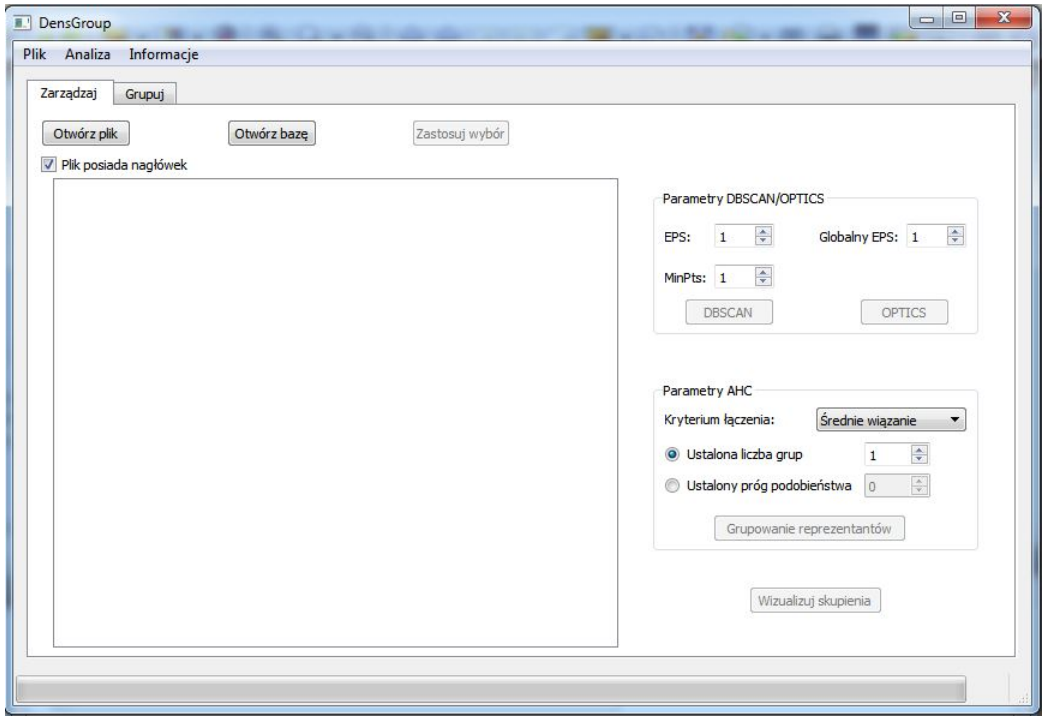
Instalacja systemu *DensGroup* ogranicza się do przekopiowania folderu z programem (znajdującego się na dołączonej do pracy płycie CD) oraz uruchomienia pliku *DensGroup.exe*. Na nośniku CD znajdują się dwie wersje oprogramowania: 32 oraz 64-bitowa. Należy zatem skopiować na dysk twardy wariant zgodny z posiadaną odmianą systemu operacyjnego. W przeciwnym przypadku program nie uruchomi się poprawnie. Dołączona do rozprawy płyta CD, zawiera prócz elektronicznej wersji pracy oraz programu, także kilka dodatkowych elementów, m.in. omawiane zbiory danych złożonych w różnych formatach. Dla przejrzystości, pełną strukturę utworzonych katalogów opisano w osobnym pliku o nazwie *struktura_katalogow.pdf*, znajdującym się w głównym folderze płyty.

7.2 Interfejs i funkcjonalność systemu DensGroup

Po uruchomieniu pliku *DensGroup.exe*, użytkownikowi prezentowany jest główny ekran programu, zilustrowany na rysunku 7.2. Interfejs systemu *DensGroup* tworzą dwie zakładki *Zarządzaj* oraz *Grupuj*. Pierwsza z nich agreguje funkcjonalności odnośnie wczytania danych (z pliku lub bazy relacyjnej), selekcji istotnych atrybutów, jak również sterowania i uruchamiania zaimplementowanych algorytmów analizy skupień oraz graficznej prezentacji ich wyników. Zakładka *Grupuj* umożliwia wyświetlenie struktury i zawartości poszczególnych skupień, czy też uzyskanie informacji o przebiegu procesu grupowania oraz podstawowych danych o konkretnej grupie (jak jej reprezentant czy liczba elementów). Dostępne są także podstawowe statystyki opisowe dla skupień jak wartości minimalne, maksymalne, średnie czy mody w odniesieniu do poszczególnych atrybutów (wchodzących w skład opisów obiektów).

Pierwszą czynnością wykonywaną przez użytkownika jest wczytanie danych do analizy (co przedstawia pozycja nr 1 na rysunku 7.3). Można tego dokonać wybierając stosowną opcję z menu głównego *Plik*, zlokalizowanego w lewej, górnej części okna lub poprzez kliknięcie na

[4]Dane do programu można wczytać z pliku tekstowego lub bezpośrednio z relacyjnej bazy danych.

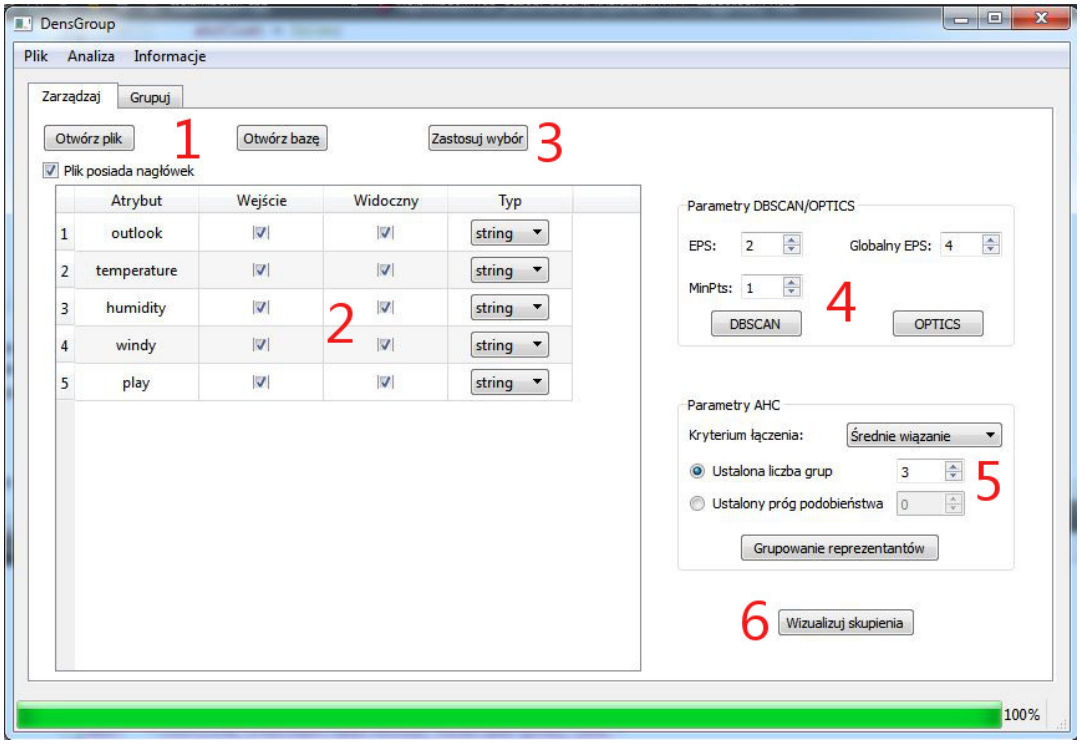


Rysunek 7.2: Główne okno systemu *DensGroup* po uruchomieniu aplikacji.
Źródło: Opracowanie własne

przycisk *Otwórz plik* czy *Otwórz bazę* w zależności od tego, czy dane zapisane są w pliku tekstowym typu CSV (ang. ***C**omma **S**eparated **V**alues*), w którym poszczególne pola rozdzielone są przecinkiem, czy też bezpośrednio z bazy danych. W pierwszym przypadku pojawia się dialog systemowy (okno), umożliwiający wybór ścieżki i określonego pliku na dysku twardym[5].

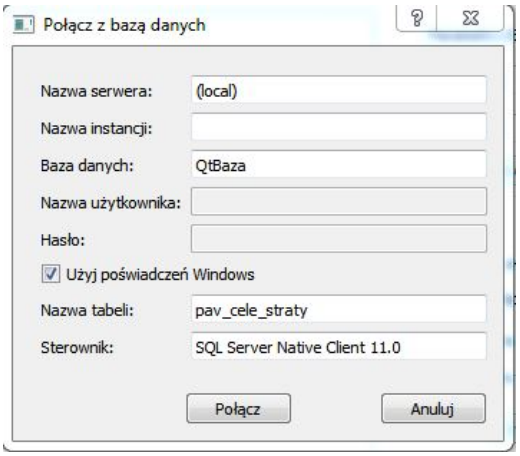
Kliknięcie na przycisk *Otwórz bazę* skutkuje natomiast wyświetleniem małego okna (zaprezentowanego na rysunku 7.4), umożliwiającego podanie adresu serwera bazy danych i innych informacji, wymaganych do zestawienia prawidłowego połączenia. Ewentualne braki sygnalizowane są stosownym komunikatem z prośbą o uzupełnienie określonych pól. Należy jednak nadmienić, że w przypadku korzystania z tej opcji, dane doczytywane są dynamicznie w miarę potrzeb. Oznacza to, że po połączeniu z bazą danych, tylko relatywnie niewielka liczba informacji jest przesyłana do programu – wyświetlane są jedynie rekordy, które mieszczą się w polu widzenia użytkownika (na zakładce *Grupuj*). Jeżeli użytkownik wykorzysta pionowy pasek przewijania, aby zobaczyć dalej położone rekordy, aplikacja skomunikuje się z bazą danych celem pobrania niezbędnych informacji. Takie podejście pozwala znacząco zredukować zapotrzebowanie na wolną pamięć RAM. Przykładowo po wczytaniu zbioru *cell_loss* w postaci pliku tekstowego system *DensGroup* zajmuje ok. 185 MB pamięci, podczas gdy wczytując ten sam zestaw danych bezpośrednio z bazy, zajętość pamięci spada do ok. 32 MB. Dzięki temu, potencjalnie można przetwarzać zbiory o większej liczności, niż gdyby wczytać je w całości do pamięci operacyjnej. Niestety jest to obarczone zwiększeniem czasu działania programu

[5]Struktura pliku wejściowego dla systemu *DensGroup* zostanie przedstawiona w dalszej części tego rozdziału.



Rysunek 7.3: Główne okno systemu *DensGroup* po procesie grupowania.
Źródło: Opracowanie własne

– narzut związany z ciągłą komunikacją z serwerem bazodanowym i doczytywaniem odpowiednich rekordów oraz zarządzaniem pamięcią, powoduje kilkukrotny spadek efektywności podczas grupowania (w stosunku do sytuacji, gdy dane wczytywane są z pliku bezpośrednio do pamięci operacyjnej). Dlatego też, jeżeli dostępna jest duża ilość pamięci RAM, zalecane jest wczytywanie danych z pliku.



Rysunek 7.4: Okno umożliwiające połączenie z bazą danych.
Źródło: Opracowanie własne

Po wczytaniu danych do programu, uzupełniana jest tabelka (pozycja nr 2 na rysunku 7.3) wskazująca atrybuty do selekcji brane pod uwagę przy procesie analizy skupień (kolumna *Wejście*). Ponadto możliwe jest pokazanie bądź ukrycie określonych kolumn na zakładce *Grupuj*[6] w zależności od ustawień, dokonanych w sekcji *Widoczny* opisywanej tabelki. Dla każdego atrybutu należy określić właściwy typ danych spośród pięciu dostępnych klas: *bool* – reprezentującą wartości zakodowane jako TRUE/FALSE (prawda/fałsz), *double* – wartości rzeczywiste, *int* – wartości w postaci liczb całkowitych, *string* – dedykowana dla atrybutów nominalnych, *uint* – reprezentująca liczby naturalne oraz zero. Przy wczytywaniu z bazy danych, typ ustalany jest automatycznie na podstawie schematu danej tabeli, jednakże użytkownik może w każdym momencie dokonać stosownej korekty. Niewłaściwe określenie typu danych może spowodować błędy konwersji oraz błędne funkcjonowanie opisywanego systemu. Na tej podstawie są bowiem wyznaczane statystyki takie jak minimum, maksimum czy średnia arytmetyczna. Następnie należy zatwierdzić dokonane zmiany w tabeli poprzez naciśnięcie przycisku *Zastosuj wybór* (pozycja nr 3 na rysunku 7.3). Dopiero wówczas odblokowane zostaną funkcjonalności grupowania gęstościowego (przypisane do przycisków *DBSCAN* lub *OPTICS*).

Kolejnym krokiem w procesie obsługi programu jest ustalenie wartości parametrów dla gęstościowych technik analizy skupień odpowiednio: w przypadku algorytmu DBSCAN – parametrów *EPS* (promienia sąsiedztwa) i *MinPts* (minimalnej liczby obiektów wchodzących w skład grupy), natomiast w przypadku algorytmu OPTICS – dodatkowo parametru *Globalny EPS* (maksymalny promień sąsiedztwa dla którego generowane jest uporządkowanie obiektów)[7]. Dla wszystkich algorytmów analizy skupień, zaimplementowano funkcję podobieństwa (lub odległości) jako liczbę cech różnych między dwoma obiektami (bądź skupieniami)[8]. Dlatego też wartość parametru *EPS* należy interpretować jako maksymalną liczbę cech, jakimi mogą różnić się dwa obiekty, aby mogły zostać potencjalnie przyporządkowane do danego skupienia. Stanowi to bezpośrednie nawiązanie do definicji tzw. odległości Hamminga [22]. Parametr *Globalny EPS* brany jest pod uwagę tylko w przypadku wykorzystywania algorytmu *OPTICS*[9]. Zgodnie z zasadą działania techniki *OPTICS*, generuje ona uporządkowanie obiektów, na podstawie którego można utworzyć ich przyporządkowanie do skupień, dla stałej wartości *MinPts* i wszystkich możliwych wartości $EPS \leqslant Globalny\ EPS$[10]. Autorzy algorytmu [4] nie podają żadnej heurystyki pomocnej przy wyznaczaniu parametru maksymalnego promienia sąsiedztwa, reprezentowanego przez *Globalny EPS* – w przytoczonym artykule jest jedynie sugestia, by omawiana wielkość była "wystarczająco duża". Dlatego też system *DensGroup*, po wczytaniu danych i zatwierdzeniu przyciskiem *Zastosuj wybór*, automatycznie ustawia wartość *Global EPS* na równą liczbie wszystkich atrybutów wejściowych. Dzięki temu po wyznaczeniu upo-

[6] Szczegóły odnośnie wyglądu zakładki *Grupuj* znajdują się w dalszej części tego rozdziału.

[7] Opis znaczenia parametrów wejściowych oraz działania zaimplementowanych algorytmów grupowania znajduje się w rozdziale 5.

[8] Ze względu na złożoność i wielkość analizowanych zbiorów danych, miara podobieństwa powinna być prosta do wyliczenia, ponieważ istotnie wpływa na czas działania algorytmów analizy skupień. Dodatkowo musi uwzględniać zarówno dane ilościowe jak i jakościowe, dlatego też ostatecznie zdecydowano się na wybór liczby cech różnych jako wspomnianą miarę.

[9] Parametr *Globalny EPS* jest odpowiednikiem odległości tworzącej (ang. *generating distance)*, występującej w opisie algorytmu w [4].

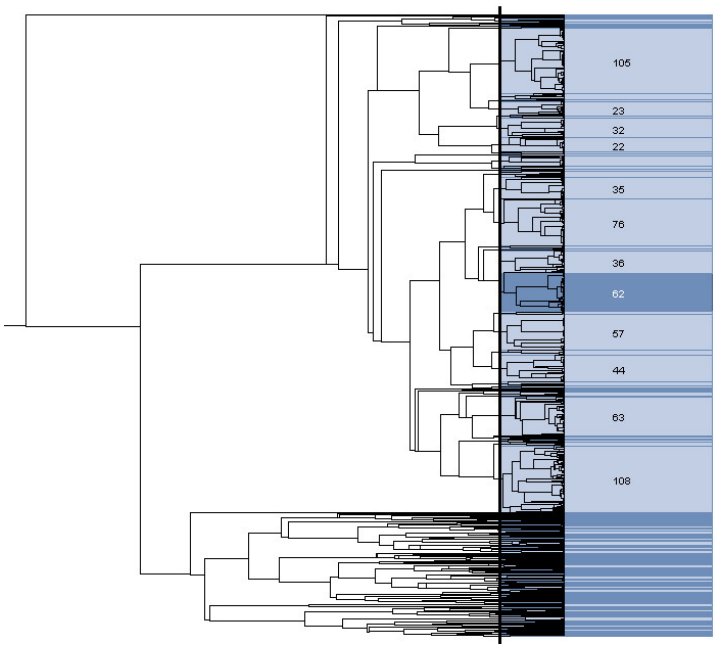
[10] Szczegóły na temat działania algorytmu *OPTICS* znajdują się w rozdziale 5 oraz [4].

rządkowania obiektów, można w szybki sposób utworzyć podział na skupienia (dla dowolnego $EPS \leqslant Globalny\ EPS$ i stałego *MinPts*).

Po ustaleniu parametrów można przystąpić do procesu grupowania gęstościowego. W tym celu należy nacisnąć przycisk *DBSCAN* lub *OPTICS* (rysunek 7.3, pozycja nr 4) w zależności od tego, jaka technika analizy skupień powinna zostać zastosowana. *DBSCAN* generuje bezpośrednio przydział do grup (który jest przedstawiany w formie listy skupień na zakładce *Grupuj*), natomiast przycisk *OPTICS* posiada dwa tryby. W pierwszym trybie generuje on uporządkowanie obiektów (na podstawie wartości parametrów *Globalny EPS* oraz *MinPts*), jak również dokonuje podziału na grupy (zgodnie z wartościami parametrów *EPS*, *MinPts*). Drugi tryb działania przycisku aktywuje się automatycznie (np. za drugim naciśnięciem) w przypadku, gdy zostało już wygenerowane uporządkowanie obiektów. Wówczas, jeżeli użytkownik modyfikował wyłącznie wartość parametru *EPS* pozostawiając resztę bez zmian (celem wygenerowania innego podziału na grupy), system *DensGroup* nie dokonuje ponownego przeliczenia uporządkowania obiektów. Pozwala to na znaczne przyspieszenie procesu generowania skupień (w stosunku do algorytmu *DBSCAN*). Postęp grupowania można śledzić na podstawie paska postępu umieszczonego na dole okna.

Wygenerowanie struktury skupień powoduje uaktywnienie się przycisków *Grupowanie reprezentantów* oraz *Wizualizuj skupienia* (zaznaczonych jako pozycje 5 i 6 na rysunku 7.3), odpowiadających kolejno za uruchomienie grupowania drugiego stopnia, ograniczonego wyłącznie do zbioru reprezentantów uprzednio wygenerowanych skupień oraz wizualizację aktualnych wyników za pomocą techniki map prostokątów (opisanej w rozdziale 6). Grupowanie reprezentantów realizowane jest za pomocą aglomeracyjnego algorytmu *AHC* [23], który w pierwszym kroku zakłada, że wszystkie obiekty ze zbioru danych stanowią odrębne skupienia, a następnie dokonuje ich iteracyjnego łączenia na podstawie wybranego kryterium. W systemie *DensGroup* zaimplementowano trzy najpopularniejsze kryteria łączenia skupień: metodę całkowitego wiązania (ang. *complete linkage*), pojedynczego wiązania (ang. *single linkage*) i średniego wiązania (ang. *average linkage*) [19]. Pozwala to na sterowanie przydziałem do grup, a w konsekwencji wpływa na ich wizualizację. Dodatkowo użytkownik może wybrać docelową liczbę skupień, jaka powinna zostać na tym etapie wygenerowana lub określić próg (nie)podobieństwa, po przekroczeniu którego hierarchiczna struktura skupień będzie przycięta.

W pierwotnej koncepcji (opublikowanej w [74]) zakładano, że algorytm AHC stworzy pełną hierarchię skupień, która będzie następnie poddawana procesowi wizualizacji. Jednakże po testach stwierdzono, że uzyskiwane drzewo hierarchiczne posiada wiele poziomów (z którego każdy reprezentuje dwa skupienia), zatem aby dotrzeć do poziomu liści (czyli poszczególnych obiektów) należałoby wielokrotnie klikać na mapie prostokątów (która ilustrowałaby zawsze jeden poziom hierarchii). W rozdziale 6 pokazano przykład mapy prostokątów wyświetlającej wszystkie poziomy hierarchii, jednakże takie podejście jest nieczytelne. Stworzenie dodatkowej wizualizacji w formie dendrogramu (po którym można nawigować klikając na interesujące poziomy hierarchii), w przypadku dużych zbiorów danych wielowymiarowych również uznano za nieefektywne. Tego typu pomysł został zaimplementowany w oprogramowaniu *Traceis Data Exploration Studio* [91]. Rysunek 7.5 przedstawia tego typu interaktywny dendrogram dla 1000 obiektów sztucznie wygenerowanego zbioru danych. Jest on szczególnie nieczytelny

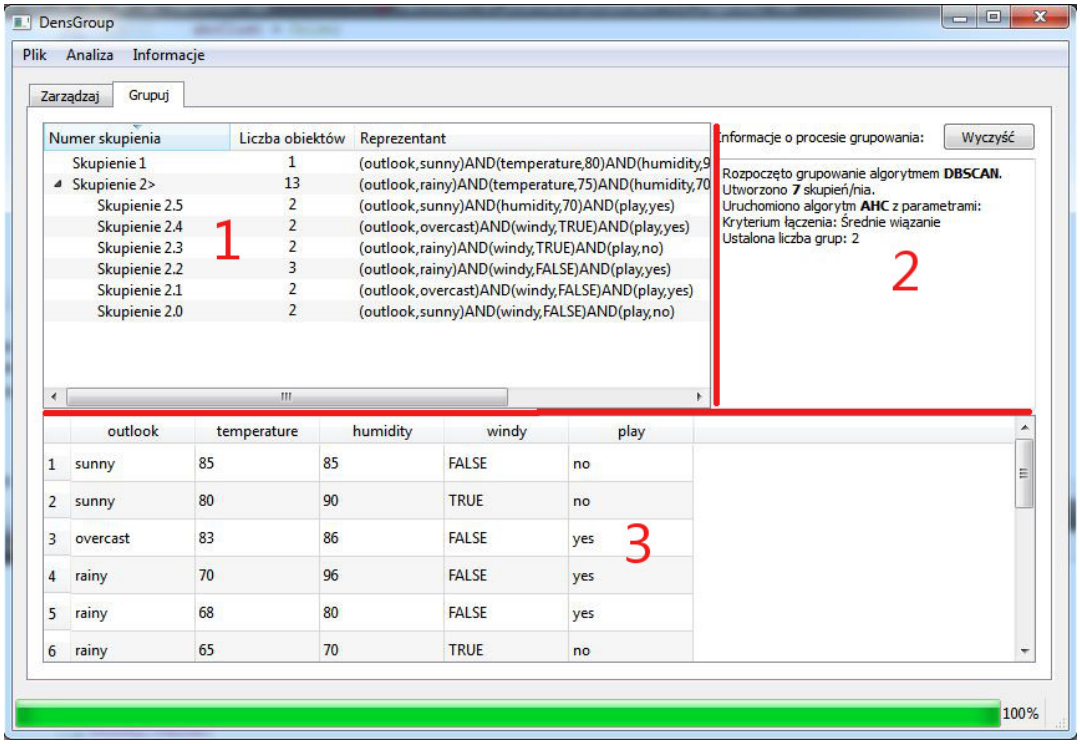


Rysunek 7.5: Dendrogram jako forma reprezentacji skupień w programie firmy Traceis.
Źródło: Opracowanie własne

w dolnych partiach dendrogramu, co uniemożliwia precyzyjne kliknięcie na interesującą grupę czy obiekt. Ponadto liczba 1000 obiektów wynika z ograniczeń programu Traceis – nie jest możliwa wizualizacja dużych zbiorów danych w tej formie. Dlatego też autor rozprawy zdecydował się wykorzystać algorytm AHC do stworzenia dwupoziomowej struktury. Jest to szczególnie przydatne w sytuacji, w której duża liczba skupień wygenerowana przez algorytm gęstościowy uniemożliwia ich efektywną wizualizację (nawet korzystając z techniki map prostokątów, która wykorzystuje całą dostępną przestrzeń roboczą). Zasada działania, zrealizowanej w systemie *DensGroup* koncepcji, jest następująca. Reprezentanci skupień poddawani są działaniu algorytmu aglomeracyjnego, aż do uzyskania pełnej struktury hierarchicznej (agregującej wszystkie bazowe skupienia w jedno). Następnie drzewo hierarchii jest przycinane do ustalonej uprzednio przez użytkownika liczby grup, bądź na podstawie zadanego progu podobieństwa. Dzięki temu użytkownik otrzymuje mniejszą liczbę skupień (wśród których pewne stanowią agregację wielu grup), którą można z łatwością poddać procesowi wizualizacji i dalszej interpretacji. Wyniki tego procesu prezentowane są również na zakładce *Grupuj*.

Wartość parametru *Próg podobieństwa* należy rozumieć jako maksymalną odległość (w sensie podobieństwa) między dwoma poziomami hierarchii, po osiągnięciu której zapamiętany jest aktualny stan drzewa i do tego poziomu zostanie ono przycięte. Jest to zatem metoda wyznaczenia liczby grup, na jaką należy podzielić bazowy zbiór skupień, ale określona za pomocą podobieństwa (na danym etapie łączenia grup). Wyłącznie od preferencji i zdolności kognitywnych użytkownika zależy, który wariant wybierze. Pozwala to jednak sterować wyglądem wizualizacji grup.

Zakładka *Grupuj* (zilustrowana na rysunku 7.6) służy do prezentacji informacji związanych

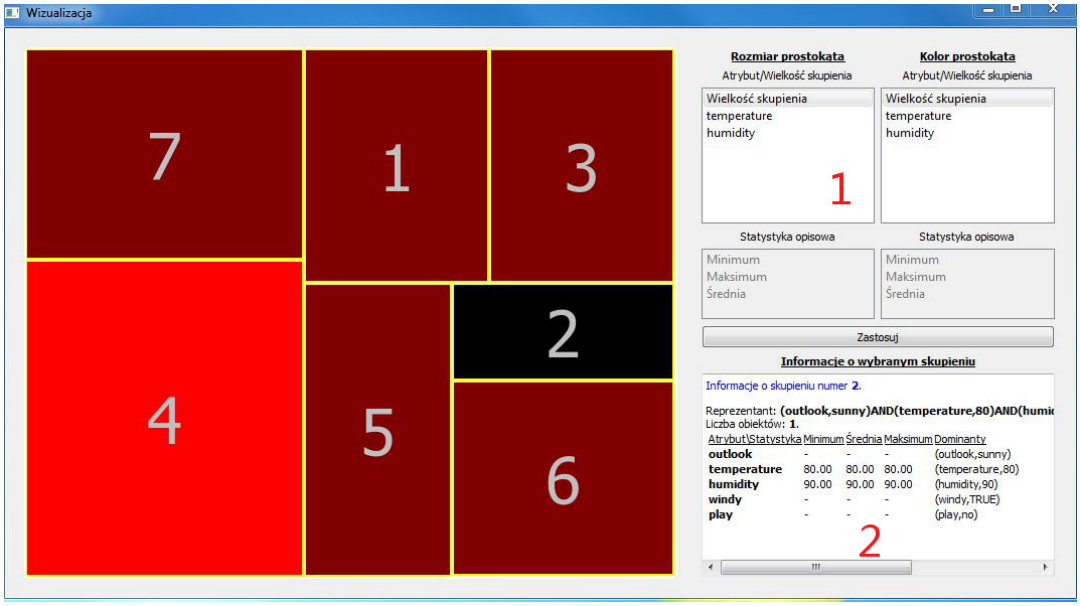


Rysunek 7.6: Wygląd zakładki *Grupuj* systemu *DensGroup*.
Źródło: Opracowanie własne

z procesem grupowania danych, takich jak przydział obiektów do skupień czy pewne statystyki opisowe. Przestrzeń górnej części zakładki podzielona jest między dwa komponenty: listę przedstawiającą hierarchiczną (bądź płaską w przypadku stosowania wyłącznie algorytmu DBSCAN) strukturę skupień (pozycja nr 1 na rysunku 7.6) oraz wielowierszowe pole tekstowe (pozycja nr 2 na rysunku 7.6), które prezentuje podstawowe informacje o procesie grupowania. Wyświetlane są: liczba wygenerowanych grup, ustawione parametry i nazwa zastosowanego algorytmu, jak również statystyki opisowe w formie wartości minimalnych, maksymalnych, średnich i mody dla atrybutów opisujących obiekty w interesującej grupie. W przypadku pierwszej listy, jeżeli skupienie agreguje kilka innych grup, w jego nazwie pojawia się znak >. Przykładowo, na rysunku 7.6 *Skupienie 2>* składa się z sześciu innych grup. Dla każdego skupienia prezentowana jest liczba obiektów w nim zawartych, jak również jego reprezentant w formie koniunkcji deskryptorów opisujących obiekty danej grupy[11]. Dzięki temu można wstępnie zbadać różnice między poszczególnymi skupieniami (na podstawie ich reprezentantów). Dodatkowo po dwukrotnym kliknięciu lewym przyciskiem myszy, prezentowane są informacje w formie statystyk opisowych dotyczące zadanego skupienia, jak również wyświetlane są wszystkie obiekty wchodzące w skład tej grupy, w postaci tabeli umieszczonej w dolnej części omawianej karty (pozycja nr *3* na rysunku 7.6). Dodatkowo użytkownik może dowolnie modyfikować rozkład komponentów na karcie *Grupuj* pociągając (z przytrzymanym lewym klawiszem myszy) za wydzielone do tego

[11]Szczegóły dotyczące zastosowanej koncepcji tworzenia opisu reprezentanta zostały zawarte w sekcji 4.3 niniejszej pracy.

sekcje (zaznaczone jako czerwone linie na rysunku 7.6). Wówczas komponenty zmieniają swój rozmiar. Przykładowo użytkownik może przeciągnąć dolną tabelę (pozycja nr 3) do góry w ten sposób, by zajęła ona całą powierzchnię karty. Jest to szczególnie przydatne w sytuacji, gdy grupa liczy wiele obiektów i użytkownik systemu chce się skupić wyłącznie na jej analizie. Zlokalizowany w prawym górnym rogu zakładki przycisk *Wyczyść* odpowiada za czyszczenie pola tekstowego wskazanego przez napis "Informacje o procesie grupowania" (pozycja nr *2* na rysunku 7.6) oraz aktualizuje dolną tabelę do stanu początkowego (czyli wyświetlającą wszystkie obiekty ze zbioru danych).



Rysunek 7.7: Wizualizacja struktury grup w systemie *DensGroup*.
Źródło: Opracowanie własne

Istotną częścią systemu *DensGroup* jest wizualizacja struktury skupień za pomocą map prostokątów, uruchamiana przez kliknięcie przycisku *Wizualizuj skupienia* (zlokalizowanego na zakładce *Zarządzaj*) lub wybór stosownego polecenia z głównego menu programu (sekcja Analiza). Wówczas tworzone jest nowe okno, którego wygląd przedstawia rysunek 7.7. Główną część okna stanowi naturalnie mapa prostokątów, gdzie każdy symbolizuje dane skupienie (zgodnie z numerem umieszczonym w środku prostokąta). Wizualizacja dostosowuje się automatycznie do aktualnej wielkości okna, przez co można ją powiększać bądź pomniejszać w zależności od dostępnego monitora i innych preferencji. Po prawej stronie okna (pozycje nr 1 i 2 na rysunku 7.7) znajduje się panel sterowania graficzną reprezentacją skupień oraz pole tekstowe wyświetlające najważniejsze informacje o aktualnie wybranej grupie. Statystyki opisujące konkretne skupienie (pozycja nr 2 na rysunku 7.7) wyświetlane są po **jednokrotnym** kliknięciu lewym przyciskiem myszy na określony prostokąt. Domyślnie wielkość prostokątów odwzorowuje liczbę obiektów w danym skupieniu. Jednakże za pomocą panelu sterowania (pozycja nr 1 na rysunku 7.7) można osobno regulować zarówno znaczenie kolorów jak również rozmiary prostokątów. Taka regulacja jest możliwa tylko biorąc pod uwagę atrybuty ilościowe (dlatego też ważne jest prawidłowe określenie typów danych, na początku interakcji z pro-

gramem). W przypadku zbioru pogodowego *weather*, na podstawie którego prezentowane są efekty działania systemu *DensGroup*, dostępne są dwa atrybuty ilościowe: temperatura (ang. *temperature*) oraz wilgotność powietrza (ang. *humidity*). Po zaznaczeniu któregokolwiek z nich, uaktywnia się lista *Statystyka opisowa* zawierająca pozycje *Minimum*, *Maksimum* i *Średnia*. Wybierając np. atrybut *temperature*, statystykę *Minimum* oraz naciskając przycisk *Zastosuj*, rozmiar bądź kolor prostokąta będzie odwzorowywał minimalne wartości temperatury w obrębie skupień. Dzięki temu użytkownik systemu może w bardzo szybki sposób zidentyfikować skupienia obserwacji o niskiej temperaturze. W zaprezentowanej na rysunku 7.7 sytuacji, zarówno wielkość jak i kolor prostokątów określają liczbę obiektów w skupieniach. Kolor ciemny oznacza niskie wartości danego parametru, natomiast jasno-czerwony wysokie. Widać wyraźnie, że skupienie numer dwa zawiera najmniejszą liczbę obiektów. W omawianym przykładzie był to tylko jeden obiekt, podczas gdy pozostałe grupy mają ich dwa lub więcej. Ponadto, jeżeli zastosowano algorytm hierarchiczny, mapa prostokątów wizualizuje właśnie wyniki tego procesu. Jeżeli prostokąt posiada w numerze znak >, symbolizuje on zawieranie innych skupień (podobnie jak miało to miejsce dla zakładki *Grupuj*). Wówczas użytkownik klikając **podwójnie** lewym przyciskiem myszy na takim prostokącie powoduje wygenerowanie wizualizacji dla niższego poziomu hierarchii. Powrót do wyższego poziomu możliwy jest poprzez **jednokrotne** kliknięcie prawym przyciskiem myszy. Pozostałe funkcje wizualizacji działają analogicznie do sytuacji przedstawionej na rysunku 7.7, gdzie przedstawiany jest płaski podział, wygenerowany przez algorytm *DBSCAN*.

7.3 Struktura plików wejściowych

System *DensGroup* przystosowany jest do pracy zarówno z relacyjnymi systemami baz danych, jak również umożliwia wczytanie informacji z pliku typu CSV. Są to pliki tekstowe, w których poszczególne wartości oddzielone są przecinkiem (bądź innym separatorem) natomiast obiekty symbolizowane są przez poszczególne linie w pliku. Struktura przykładowego pliku CSV, rozpoznawanego przez system *DensGroup* prezentuje się następująco:

```
outlook,temperature,humidity,windy,play
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no
```

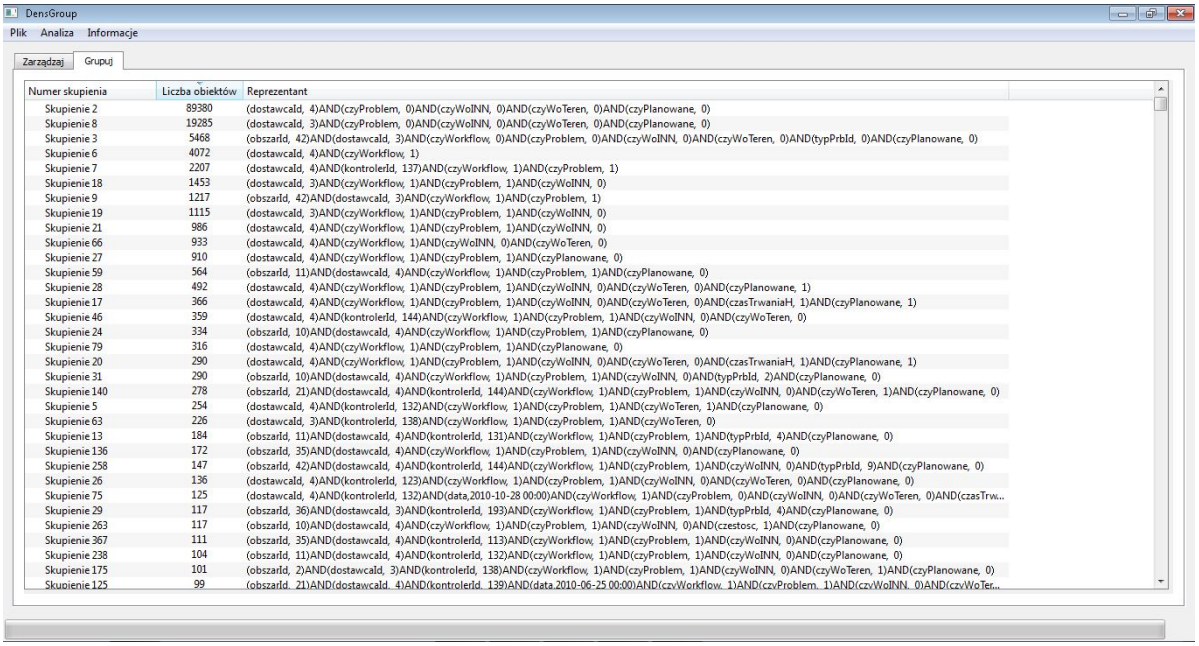
W prezentowanym przykładzie plik w pierwszej linii zawiera nagłówek, czyli określenie nazw poszczególnych kolumn (uzyskanych przez odseparowanie wartości w jednej linii przecinkiem), reprezentujących atrybuty. Wówczas należy w systemie *DensGroup* zaznaczyć opcję *Plik posiada nagłówek* (zlokalizowaną na zakładce *Zarządzaj*) przed jego wczytaniem. Nazwy kolumn bądź wartości kategoryczne nie powinny zawierać spacji. Ponadto wartości rzeczywiste muszą być zapisane z kropką (np. 26.5 zamiast 26,5) jako separatorem.

Plik o podobnej do podanej strukturze, powinien mieć rozszerzenie **.data*, **.txt* lub **.csv*. Ma to jednak wyłącznie wpływ na filtrowanie danych, w systemowym oknie wyboru pliku do otwarcia (dzięki czemu widoczne są tylko te w podanych formatach). Rozszerzenie **.data* wykorzystywane jest przez zestawy danych, zgromadzone w *UC Irvine Machine Learning Repository*, przez co można w łatwy sposób importować je do programu. Należy jednak pamiętać, że pliki z repozytorium nie posiadają najczęściej nagłówka. W takiej sytuacji poszczególne atrybuty oznaczane są poszczególnymi liczbami naturalnymi. Należy również wspomnieć, że system *DensGroup* umożliwia import wyników analizy skupień wygenerowanych w programie Weka (poprzez zastosowanie algorytmu DBSCAN lub OPTICS). W tym celu należy przekopiować całą zawartość okna programu Weka, po zakończeniu działania wspomnianych algorytmów, do pliku tekstowego oraz wybrać stosowną opcję z menu głównego Plik. Pozwala to na kontynuowanie analizy danych, rozpoczętej przy użyciu tego popularnego programu.

7.4 Wizualizacja skupień dla zbioru *cell_loss* przy użyciu narzędzia DensGroup

Algorytmy analizy skupień mają na celu pogrupowanie źródłowego zbioru danych, w celu wykrycia nowych i potencjalnie użytecznych zależności między poszczególnymi obiektami w sposób nienadzorowany, czyli bez pomocy użytkownika lub dostarczania zewnętrznej wiedzy (jak na przykład zestawu klas, na jakie należy podzielić dane). Jednakże w kontekście analizy rzeczywistych danych złożonych, problematyczna staje się interpretacja uzyskanych wyników (w postaci struktury grup), nawet uwzględniając wyłącznie przegląd wygenerowanych reprezentantów skupień. Literatura przedmiotu często zakłada określoną relację, między liczbą obiektów ze zbioru danych (ozn. N) oraz liczbą stworzonych skupień (ozn. lSk), według której $lSk << N$ [22]. Dzięki temu utworzona liczba grup jest relatywnie niewielka, przez co możliwe jest ich porównanie w rozsądnym czasie wykorzystując wrodzone zdolności kognitywne człowieka. Niestety w rzeczywistych zastosowaniach (np. segmentacji klientów sieci hipermarketów) liczba obiektów mierzona jest w setkach tysięcy bądź milionach, przez co również algorytmy analizy skupień mogą wytworzyć dużą liczbę grup (rzędu kilku tysięcy bądź większą). Tego typu struktura jest bardzo trudna w analizie (szczególnie porównawczej) w rozsądnym czasie. Dlatego też obecnie poszukuje się rozwiązań przedstawionego problemu, a jednym z nich może być, proponowane przez autora niniejszej rozprawy, zastosowanie podwójnego grupowania oraz wizualizacji wyników w formie map prostokątów, jak to zostało zrealizowane w systemie *DensGroup*. Celem niniejszej sekcji jest krótkie zapoznanie z problematyką zbyt dużej liczby grup (do skutecznej interpretacji), na przykładzie rzeczywistego zbioru *cell_loss* (agregującego informacje o nieprawidłowościach pracy urządzeń nadawczo-odbiorczych) oraz rozwiązanie

tego problemu przy użyciu narzędzia *DensGroup*.

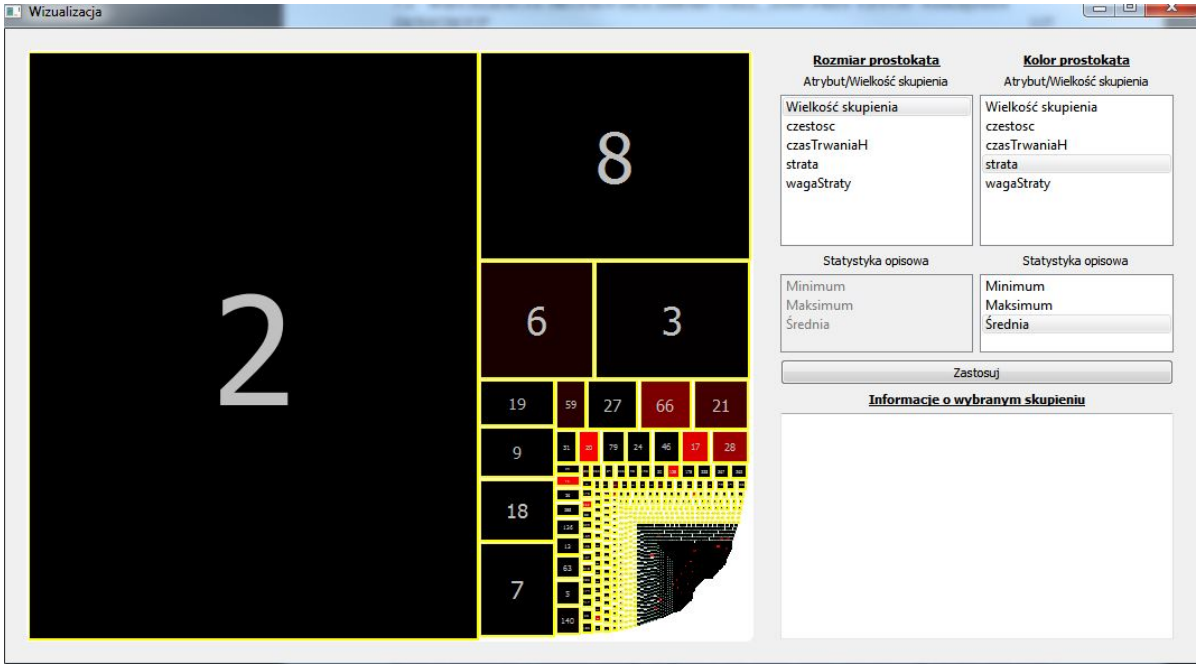


Rysunek 7.8: Przykładowy wynik działania algorytmu *DBSCAN* dla zbioru *cell_loss*.
Źródło: Opracowanie własne

Rysunek numer 7.8 przedstawia wynik działania gęstościowego algorytmu *DBSCAN* dla zbioru *cell_loss* (i parametrów wejściowych $Eps = 2$, $MinPts = 1$). Zostało wygenerowanych 1869 skupień, liczących od jednego do niemalże dziewięćdziesięciu tysięcy obiektów. Przypadek ten może być uznawany za zbyt skomplikowaną strukturę do skutecznej i szybkiej analizy przez człowieka, a to właśnie na użytkowniku systemu wydobywania wiedzy, najczęściej spoczywa obowiązek interpretacji otrzymanych wyników, jak również formułowania i potwierdzenia wniosków badawczych. Nawet użycie wizualizacji w formie map prostokątów, celem wykorzystania zdolności kognitywnych człowieka przy analizie dużych zbiorów danych, okazuje się w takiej sytuacji niewystarczające. Rysunek 7.9 przedstawia wspomnianą, graficzną prezentację struktury grup. Niestety małoliczne grupy są niemalże niewidoczne na wizualizacji. Poprawa tego stanu rzeczy wymagałaby zastosowania większego ekranu, zniekształcenia wizualizacji lub wykorzystania przybliżenia. Jednakże żadne z tych rozwiązań nie jest w opinii autora satysfakcjonujące. Ponadto występuje problem z właściwym rozmieszczeniem prostokątów w prawym, dolnym rogu ekranu. Wynika to z faktu, że mapy prostokątów zajmują dokładnie cały obszar roboczy, tylko w przypadku, gdy suma pól wszystkich prostokątów jest równa polu powierzchni obszaru roboczego (lub jego wielokrotności)[12].

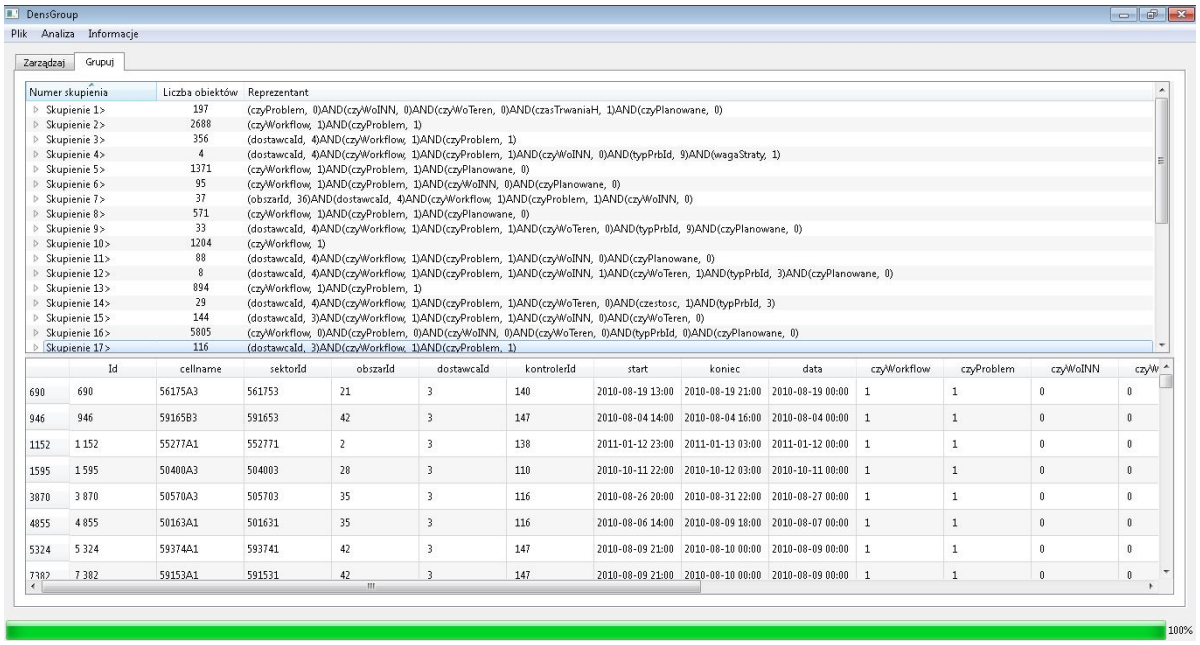
Dlatego też proponuje się, aby utworzoną strukturę grup poddać kolejnemu procesowi analizy skupień, jednakże tym razem wykorzystując inny algorytm np. z grupy aglomeracyjnych,

[12]Czasami elementy mapy prostokątów są sztucznie zniekształcane oraz dopasowywane są do nich odpowiednie odstępy, by uzyskać efekt wypełnienia całego obszaru roboczego. Jednakże każde zniekształcenie wizualizacji może doprowadzić do błędnych wniosków na temat reprezentowanej przez nią struktury grup, dlatego też autor nie zdecydował się na wprowadzenie tego typu modyfikacji.



Rysunek 7.9: Wizualizacja struktury skupień wygenerowanych przez algorytm *DBSCAN* dla zbioru *cell_loss*.

Źródło: Opracowanie własne



Rysunek 7.10: Przykładowy wynik zastosowania algorytmu *AHC* na zbiorze skupień komórek.

Źródło: Opracowanie własne

celem uzyskania dwupoziomowej hierarchii. Wówczas reprezentantów utworzonych skupień, należy interpretować jako wzorce stanowiące uogólnienie powiązań i regularności występujących niejawnie w danych. Dla tychże wzorców uruchamiany jest algorytm AHC, który ze względu na rozmiar analizowanych danych i ich złożoną strukturę, nie mógł zostać wcześniej bezpośrednio wykorzystany. Jako kryterium łączenia skupień, zastosowano metodę średniego wiązania. Rysunek 7.10 przedstawia wynik opisywanego kroku, dla arbitralnie ustalonej liczby 30 grup (uznanej przez użytkownika jako dostatecznie małolicznej w celu dalszej interpretacji). Pozwala to na znaczne ograniczenie liczby grup, które mogą zostać przebadane w rozsądnym czasie (na podstawie opisu reprezentanta, bądź zawartości danego skupienia i prostych statystyk opisowych, omówionych w poprzedniej sekcji). Wpływa to również bardzo korzystnie na wygląd samej wizualizacji skupień, co przedstawiono na rysunku 7.11.

Rysunek 7.11: Przykładowa mapa prostokątów dla zbioru skupień komórek, po zastosowaniu algorytmu aglomeracyjnego.

Źródło: Opracowanie własne

Rysunek 7.11 prezentuje mapę prostokątów, w której wielkość elementów odpowiada średniej częstości występowania zarejestrowanych zdarzeń (związanych z niedostępnością poszczególnych komórek) w ramach danego skupienia, natomiast kolor odwzorowuje średni czas trwania określonego zdarzenia, mierzony w godzinach. Im większe pole danego prostokąta, tym więcej zdarzeń będących wynikiem niedostępności konkretnych urządzeń nadawczo-odbiorczych, zapisano w tabeli danych. Podobnie im bardziej czerwony kolor prostokąta, tym dłużej trwały te zdarzenia. Na podstawie prezentowanej wizualizacji można stwierdzić, że generalnie zdarzenia mają niski czas trwania (rzędu jednej godziny), co symbolizuje przewaga ciemnego koloru. Wyjątkiem są jednak skupienia numer *14>*, *29>* oraz *3>*. Po kliknięciu na skupienie czternaste, otrzymano kilka informacji o tej grupie (przedstawionych w prawym dolnym rogu rysunku 7.11), z których wynika, że zawiera ona przynajmniej jedno urządzenie nadawczo-

odbiorcze, które miało problem z dostępnością przez 261 godzin. Tak długi czas niedostępności świadczy o celowym wyłączeniu danej komórki bądź grupy komórek, co zostało potwierdzone przez eksperta dziedzinowego. Dzięki relatywnie prostemu rozwiązaniu, uzyskano konkretną wiedzę (poprzez identyfikację urządzenia znacząco odstającego od reszty, pod kątem niedostępności), która nie była bezpośrednio zauważalna wcześniej, ze względu na nadmiar danych (o strukturze skupień) do przetworzenia. Potwierdza to przydatność proponowanej koncepcji i autorskiego systemu *DensGroup* w zadaniu analizy danych złożonych.

7.5 Podsumowanie

Niniejszy rozdział przedstawia motywację do stworzenia autorskiego systemu wydobywania wiedzy z danych złożonych o nazwie *DensGroup*, jego projekt i szczegóły implementacyjne. Opisuje on wszystkie funkcjonalności programu, celem zapoznania użytkownika końcowego z jego obsługą. Wyszczególnione są również wymagania sprzętowe aplikacji oraz sposób jej instalacji. Istotnym elementem tego rozdziału jest również sekcja opisująca działanie dwuetapowego grupowania (stosowanego zarówno do obiektów zbioru wejściowego jak również ograniczonego wyłącznie do reprezentantów skupień), jako techniki pozwalającej na wizualizację i interpretację dużych zbiorów danych wielowymiarowych.

Rozdział 8

Eksperymenty obliczeniowe

Przeprowadzone w ramach niniejszej rozprawy eksperymenty mają na celu pokazanie przebiegu procesu odkrywania wiedzy z danych złożonych, jak również potwierdzenie słuszności zastosowania w tym celu technik analizy skupień. Grupowanie zostanie przeprowadzone korzystając zarówno ze stworzonego w celu ekstrakcji wiedzy systemu *DensGroup* (który wykorzystuje algorytmy gęstościowe i hierarchiczny), jak również poprzez wygenerowanie stosownych zapytań języka SQL (ponieważ dane źródłowe przechowywane są w relacyjnej bazie danych). Należy nadmienić, że wszystkie analizy oparte są o dwa rzeczywiste zestawy danych wielowymiarowych, dotyczących zagadnień telekomunikacyjnych i sieciowych (które zostały szczegółowo opisane w rozdziale 2). Dlatego też przeprowadzone badania zostały podzielone na dwie osobne sekcje: dotyczące zestawu *cell_loss* oraz *ap_loss*. Wnioski z zamieszczonych w pracy eksperymentów zostały również potwierdzone przez ekspertów dziedzinowych[1]. Dzięki temu zweryfikowano poprawność oraz użyteczność odkrytej wiedzy (wyrażonej przykładowo w formie korelacji między danymi urządzeniami), a tym samym możliwość jej bezpośredniego zastosowania w wykorzystywanych rozwiązaniach monitorujących działanie sieci bądź przy planowaniu zmian (w strukturze i rozmieszczeniu urządzeń bądź łączy komunikacyjnych).

Wykaz eksperymentów przeprowadzonych w ramach niniejszej pracy jest następujący:

- **Eksperymenty 1, 5**: Analiza przydatności wykorzystania histogramów, jako metody opisu danych, w procesie odkrywania wiedzy.
- **Eksperymenty 2, 6**: Analiza możliwości zastosowania klauzul grupujących i funkcji agregujących języka SQL, celem znalezienia korelacji między obiektami (bądź atrybutami) zbiorów złożonych.
- **Eksperymenty 3, 7**: Wyznaczenie optymalnych parametrów sterujących dla gęstościowych algorytmów grupowania.

[1]Pojęcie eksperta dziedzinowego jest tutaj utożsamiane z operatorem danej bazy, który w ramach swojej pracy również analizował wspomniane zestawy danych, jednakże innymi metodami niż zaprezentowane w rozprawie. Należy również nadmienić, że przedstawione w ramach rozprawy wnioski i zależności nie zostały wcześniej wykryte przy wspomnianych analizach eksperckich.

- **Eksperymenty 4, 8**: Analiza przydatności zastosowania systemu *DensGroup* w procesie odkrywania wiedzy.

8.1 Wydobywanie wiedzy ze zbioru *cell_loss*

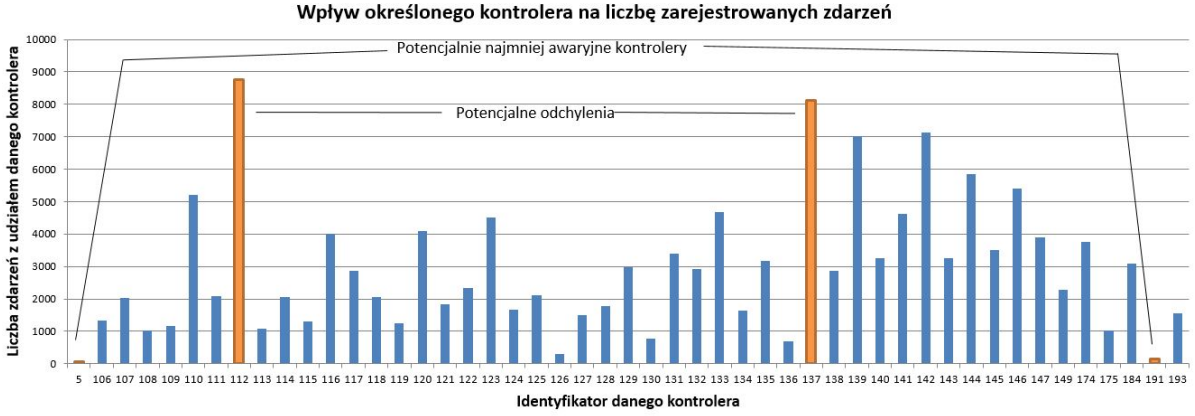
Pierwszym krokiem procesu wydobywania wiedzy jest zapoznanie się ze strukturą danych. Zgodnie z zasadami eksploracyjnej analizy danych (omówionej szczegółowo w rozdziale 4), zdecydowano się na stworzenie szeregu wykresów częstości występowania wartości unikalnych dla wszystkich atrybutów wchodzących w skład opisów obiektów zbioru *cell_loss*[2].

Eksperyment 1. Analiza przydatności wykorzystania histogramów, jako metody opisu danych, w procesie odkrywania wiedzy dla zbioru cell_loss

Dane potrzebne do stworzenia histogramów uzyskano korzystając wyłącznie z możliwości silnika bazodanowego, formułując następujące zapytanie SQL:

```
select @atr, count(@atr) as liczba from cell_loss group by @atr
```

gdzie `atr` to zmienna przechowująca nazwę atrybutu, dla którego mają zostać wyznaczone (przyjmowane przez niego) wartości unikalne oraz ich liczba, w obrębie badanego zestawu danych. Potencjalnie pozwoli to na identyfikację urządzeń nadawczo-odbiorczych, odstających od innych pod kątem określonych parametrów. Przykładowo, wykres zilustrowany na rysunku 8.1 pozwala ocenić wpływ zastosowania określonego kontrolera na liczbę zarejestrowanych zdarzeń[3].



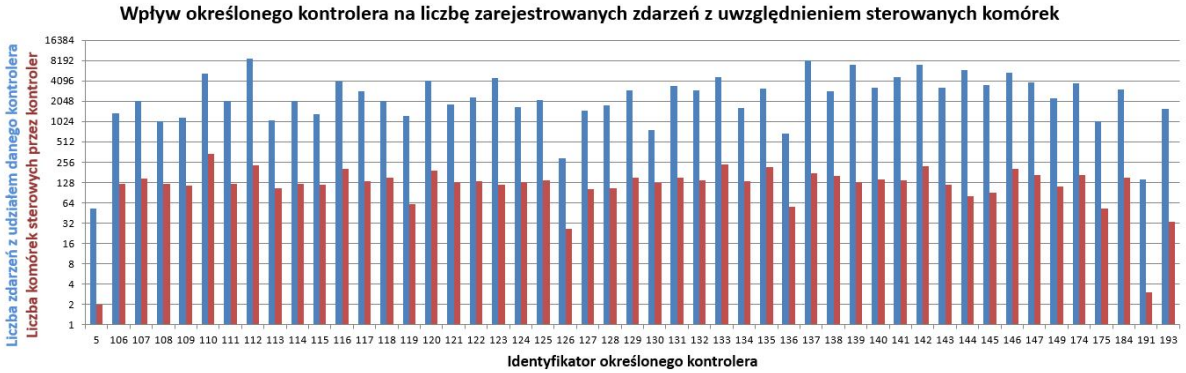
Rysunek 8.1: Wpływ określonego kontrolera na liczbę zarejestrowanych zdarzeń.
Źródło: Opracowanie własne

Analizując wspomniany wykres można dojść do wniosku, że kontrolery o numerach 112 oraz 137 stanowią odchylenia, ponieważ jako jedyne biorą one udział w ponad ośmiu tysiącach

[2]W niniejszej pracy zamieszczono jedynie najistotniejsze, z punktu widzenia analizy danych, wykresy i wnioski.

[3]Zarejestrowana liczba zdarzeń jest porównywalna z liczbą obiektów zapisanych w bazie.

zdarzeń. Dodatkowo można by stwierdzić, że najmniej problemów związanych z dostępnością usług sieciowych, sprawiają kontrolery o numerach 5 oraz 191, ponieważ odpowiednio 53 razy oraz 140 razy zostały zarejestrowane zdarzenia z ich udziałem, co dla ponad 140 tysięcy wpisów w badanym zbiorze jest pomijalną wartością. Niestety tego typu analiza jest błędna. Wynika to z faktu, że pod kontrolą danego kontrolera pracuje zwykle od kilku do kilkudziesięciu komórek. Oczywistym jest, że im więcej urządzeń nadawczo-odbiorczych przypisanych jest do określonego kontrolera, tym większe prawdopodobieństwo, że przynajmniej jedna z nich może być niedostępna, przez co identyfikator takiego kontrolera częściej będzie pojawiał się w statystykach. Dlatego też na wykresie 8.2, uwzględniono również liczbę sterowanych komórek oraz zastosowano skalę logarytmiczną na osi rzędnych (celem poprawy czytelności).



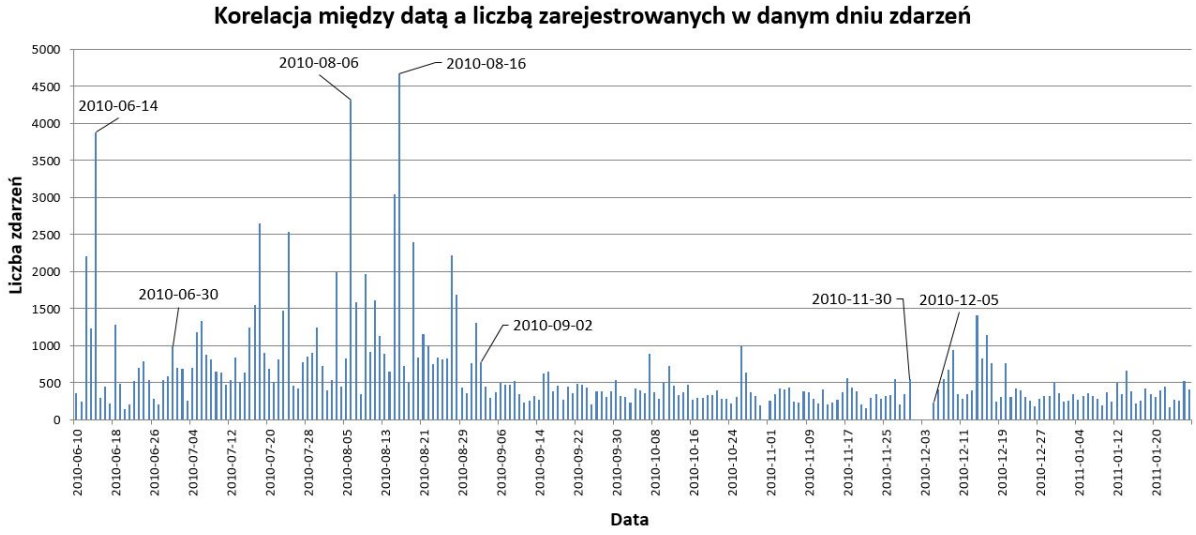
Rysunek 8.2: Wpływ określonego kontrolera na liczbę zarejestrowanych zdarzeń z uwzględnieniem sterowanych komórek.

Źródło: Opracowanie własne

Tym razem z rysunku 8.2 można wywnioskować, że choć zidentyfikowane wcześniej kontrolery o numerach 112 oraz 137 biorą udział w największej liczbie zarejestrowanych zdarzeń, to jednak odpowiadają one również za sterowanie znacznie większą liczbą komórek niż kontrolery 5 i 191, pojawiające się najrzadziej w zbiorze danych. Nie sposób zatem porównać pracy kontrolerów między sobą, analizując wyłącznie częstość ich występowania w zbiorze *cell_loss*. W dalszej części niniejszej pracy zostanie przedstawiony inny sposób identyfikacji potencjalnie problematycznych kontrolerów.

Biorąc pod uwagę schemat bazy *cell_loss*, kolejnym interesującym (z punktu widzenia analityka) atrybutem, może być data zarejestrowania zdarzenia. Wykres 8.3 przedstawia korelację między datą a liczbą zarejestrowanych w danym dniu zdarzeń. Pozwala to wizualnie rozpoznać, w jakich okresach czasu miało miejsce najwięcej zdarzeń związanych z pracą całej sieci telekomunikacyjnej. Już na pierwszy rzut oka widać, że najbardziej wyróżniającym się (pod kątem niedostępności komórek) okresem czasu, jest ten między 30 czerwca a 2 września, ze szczególnie zaznaczonymi maksymami lokalnymi w dniach szósty oraz szesnasty sierpnia. Może to być spowodowane wysokimi temperaturami występującymi podczas okresu wakacyjnego – upały powodują przegrzewanie się urządzeń sieciowych. Jednakże w tym przypadku, w miesiącach lipcu i sierpniu 2010 roku, występowały gwałtowne burze, ulewne deszcze i porywiste wiatry nad całym obszarem kraju. Szczególnie silne wyładowania atmosferyczne zarejestrowa-

no w dniach 6 oraz 16 sierpnia, co tłumaczy wykazane ogromne straty dostępności i problemy w działaniu urządzeń sieciowych. Znacznie bardziej interesujące są dwa okresy czasu: 14 czerwca oraz przedział między 1 i 4 grudnia 2010 roku.



Rysunek 8.3: Korelacja między datą a liczbą zarejestrowanych w danym dniu zdarzeń.
Źródło: Opracowanie własne

W pierwszym przypadku dnia 14 czerwca 2010 roku zarejestrowano 3881 zdarzeń związanych z pracą komórek. Brało w nich udział ponad 40% wszystkich urządzeń nadawczo-odbiorczych zapisanych w zbiorze *cell_loss*, pracujących pod kontrolą 46 różnych kontrolerów (z 49 możliwych) oraz rozlokowanych we wszystkich 10 różnych obszarach geograficznych. Powodem tych zdarzeń są ponownie niekorzystne warunki pogodowe – w dniach 13 oraz 14 czerwca odnotowano ulewne deszcze i wezbrały rzeki, co tłumaczy dużą liczbę interwencji zespołu do spraw utrzymania sieci (napraw) w związku z wykrytymi stratami dostępności. Z całą pewnością automatyczne identyfikowanie tego typu zdarzeń i dni, na przestrzeni dłuższego przedziału czasu (np. w skali roku), powinno pozwolić operatorowi sieci na dalszą analizę i ewentualne wdrożenie rozwiązań temu zapobiegających.

Kolejny interesujący okres czasu to dni między 1 i 4 grudnia 2010 roku, ponieważ wówczas system monitorujący nie zarejestrował żadnych zdarzeń związanych z niedostępnością komórek. Jest to oczywiście bardzo mało prawdopodobne, by wszystkie urządzenia pracowały z pełną wydajnością, dlatego też sugeruje to celowe wyłączenie monitorowania ich pracy, bądź awarię systemu gromadzenia danych. Ekspert dziedzinowy potwierdził, że powodem wykrytej anomalii była niewłaściwa praca systemu gromadzenia informacji.

Pozostałe wygenerowane wykresy nie przyczyniły się do odkrycia nowej wiedzy na temat analizowanego zestawu danych. Pierwotnie interesujący wydawał się rozkład sektorów i obszaru geograficznego, w jakim występują problemy z dostępnością komórek, jednakże podobnie jak w przypadku problemu częstości występowania kontrolerów (opisanym na początku tej sekcji), nie można tych danych analizować bezpośrednio. Jak zaznaczono bowiem w rozdziale 2, na pracę konkretnego urządzenia nadawczo-odbiorczego silny wpływ mają panujące warun-

ki atmosferyczne, ukształtowanie terenu i konfiguracja budynków czy ulic, przez co bez tych informacji nie da się porównać pracy komórek rozlokowanych w różnych częściach badanego obszaru. Pozostałe stworzone wykresy potwierdzają jedynie znane fakty, że większość zaistniałych zdarzeń nie była zaplanowana wcześniej, zdarzenia powodowały relatywnie niedużą stratę (czyli komórka była niedostępną jedynie przez niewielki okres czasu w ciągu godziny) oraz znaczna część urządzeń pochodzi od tego samego dostawcy.

Eksperyment 2. Analiza możliwości zastosowania klauzul grupujących i funkcji agregujących języka SQL, celem znalezienia korelacji między obiektami (bądź atrybutami) zbioru cell_loss

Kolejna część eksperymentów skupiała się na wydobywaniu wiedzy poprzez wykonywanie grupowania bezpośrednio na serwerze bazy danych, za pomocą odpowiednio sformułowanych poleceń języka SQL. Typowo taka agregacja dokonywana jest za pomocą klauzuli *group by*, jednakże nowsze wersje Microsoft SQL Server udostępniają klauzulę analityczną *over*. Mimo podobieństw, ponieważ oba zapisy umożliwiają korzystanie z funkcji agregujących (pozwalających wyznaczyć minimum, maksimum, liczność czy średnią dla interesujących atrybutów wchodzących w skład opisów obiektów należących do jednej grupy), posiadają one bardzo istotną różnicę, wpływającą na możliwości i czas działania zapytania. Mianowicie klauzula *group by* najpierw dokonuje grupowania na podstawie podanych kryteriów i zwraca po jednym wyniku (rekordzie) dla każdej grupy. Powoduje to ograniczenie, że można odwoływać się wyłącznie do kolumn, według których dane zostały pogrupowane (czyli zawartych w klauzuli *group by*). Natomiast klauzula *over* z argumentem *partition by*, pozwala na skorzystanie z wymienionych wcześniej funkcji agregujących, jednocześnie zwracając pełen zestaw rekordów (a nie tylko jeden dla każdej grupy). Można to przyrównać do tworzenia tabeli tymczasowej, w której dane podzielone są na konkretne grupy i na podstawie tych skupień wyznaczana jest konkretna wartość (np. średnia arytmetyczna), która zostaje potem uwzględniona w rezultacie bazowego zapytania. Pozwala to przykładowo na zliczenie elementów w ramach danej grupy, jak również ich wyświetlenie w jednym zapytaniu (co nie byłoby możliwe korzystając wyłącznie z *group by*). Dodatkowo w wersji 2012 SQL serwera można wykorzystać tzw. mechanizm ramek (ang. *frames*), dzięki czemu można wyznaczyć podzbiór z innego podzbioru obiektów. Umożliwia to wyznaczanie sum narastających lub innych agregacji, dotychczas realizowanych za pomocą podzapytań lub zapytań skorelowanych, co upraszcza powstały kod SQL, jak również pozytywnie wpływa na efektywność takich zapytań (ponieważ są one lepiej zoptymalizowane przez algorytmy serwera bazodanowego)[4].

Celem pierwszego zapytania SQL jest wyznaczenie tych urządzeń nadawczo-odbiorczych (wraz z ich właściwościami[5]), które w tym samym czasie, miały dokładnie taką samą stratę (poziom dostępności). Pozwoli to na powiązanie urządzeń, przypisanych przez system moni-

[4]Szczegóły na temat składni i działania klauzul *group by* i *over* znajdują się w oficjalnej dokumentacji SQL Server 2012, dostępnej pod adresami http://technet.microsoft.com/pl-pl/library/ms189461.aspx oraz http://technet.microsoft.com/pl-pl/library/ms177673.aspx.

[5]Kolumny zawarte w klauzurze *select* mogą być dowolne, jednakże zostały wybrane zdaniem autora tylko najważniejsze z punktu widzenia danego zapytania. Dla pierwszego zapytania istotne było dołączenie m.in. kolumny *zdarzenieId* by pokazać, że udało się skorelować potencjalnie różne zdarzenia.

torujący do innego zdarzenia (o innym identyfikatorze), jednakże w rzeczywistości, będących częścią tej samej awarii. Przykładowo, jeżeli grupa komórek (pozornie niezwiązanych ze sobą) charakteryzuje się w tym samym punkcie czasowym, dokładnie takim samym stopniem dostępności, może to sugerować problem z kontrolerem, który nimi steruje. Sformułowano zatem zapytanie SQL, przedstawione na listingu 8.1, które potencjalnie pozwoli na wykrycie problemów z funkcjonowaniem kontrolerów komórek.

Listing 8.1: Komórki o tym samym poziomie dostępności.

```
select liczebnosc, cellname, obszarId, dostawcaId, kontrolerId, zdarzenieId,
round(strata, 7) as strata
from (
  select count(*) over (partition by start, koniec, strata) as liczebnosc,
  t.*
  from cell_loss t
  where t.[data] = '07-08-2010'
) t1
where t1.liczebnosc >= 10
```

Pytanie zapisane na listingu 8.1 najpierw zlicza ile komórek miało dokładnie taką samą dostępność w tym samym oknie czasowym (równym godzinie, bo w takich odstępach zapisywane są pomiary) siódmego sierpnia 2010 (przechowując tę wartość w polu *liczebnosc*), a następnie wyznacza z tego zbioru rekordy, których liczebność jest co najmniej równa dziesięć. Data *07-08-2010* została wybrana arbitralnie, aby ograniczyć zbiór wynikowy (dla czytelności), jednakże może to być dowolny inny okres czasu (np. miesiąc czy więcej). Należy nadmienić, że ograniczenie do konkretnego dnia pozwala analitykowi prezentować statystyki dostępności na bieżąco (dzień po dniu). Warunek *t1.liczebnosc >= 10* pozwala sterować "czułością" wykrywania – przykładowo dla operatora sieci istotne mogą być wyłącznie zdarzenia, które dotyczą większej liczby komórek. Sytuacje, które odnoszą się do dwóch, trzech komórek powinny być wiązane z kontrolerem, ponieważ zazwyczaj steruje on pracą większej liczby urządzeń. Rezultat opisywanego zapytania ilustruje tabela 8.1.

Zapytanie 8.1 zwraca dwa skupienia liczące odpowiednio 12 oraz 15 rekordów (na podstawie kolumny *liczebnosc*). Analizując zawartość obu grup można stwierdzić, że dotyczą one urządzeń tego samego producenta, pracujących na tych samych obszarach geograficznych i posiadających dokładnie tę samą stratę, wynoszącą niecałe 3% (co oznacza, że w ciągu godziny pomiarowej, komórki w tych skupieniach były niedostępne przez prawie 2 minuty). W przypadku pierwszego skupienia, komórki są sterowane przez kontrolery 116 lub 138, natomiast w przypadku drugiej grupy są to kontrolery numer 138 i 110. Ponieważ w ramach danego skupienia występuje więcej niż jeden kontroler, nie można stwierdzić, że niedostępność urządzeń nadawczo-odbiorczych wynika z awarii kontrolera. Dlatego też w tym przypadku, najbardziej prawdopodobną przyczyną niedostępności komórek jest awaria radiolinii (np. w skutek chwilowej przerwy w zasilaniu). Jednakże samo zapytanie 8.1 ma jedną wadę: stwierdza, że przyczyną problemu jest radiolinia, lub kontroler. Jest to dość nieprecyzyjne, szczególnie w przypadku, gdyby tego typu metoda wykrywania przyczyny awarii, miałaby zostać zautomatyzowana i wykorzystana jako moduł większego systemu monitorującego. Dlatego też wspomniane zapytanie 8.1 zostało przez autora

Tabela 8.1: Wynik zapytania 8.1

liczebnosc	cellname	obszarId	dostawcaId	kontrolerId	zdarzenieId	strata
12	50132A2	35	3	116	34114	0,0277778
12	55277B1	2	3	138	409556	0,0277778
12	50132B1	35	3	116	348203	0,0277778
12	55277B3	2	3	138	835073	0,0277778
12	55277B2	2	3	138	616176	0,0277778
12	50132A3	35	3	116	523154	0,0277778
12	55277A2	2	3	138	693381	0,0277778
12	50132A1	35	3	116	280009	0,0277778
12	50132B3	35	3	116	677842	0,0277778
12	50132B2	35	3	116	117526	0,0277778
12	55277A3	2	3	138	919462	0,0277778
12	55277A1	2	3	138	480253	0,0277778
15	50990B1	35	3	110	812947	0,0277778
15	55164A1	2	3	138	234199	0,0277778
15	55164A2	2	3	138	921761	0,0277778
15	55164B1	2	3	138	369401	0,0277778
15	50264B2	35	3	110	15825	0,0277778
15	50264B1	35	3	110	657815	0,0277778
15	50990A1	35	3	110	35467	0,0277778
15	50990A3	35	3	110	49839	0,0277778
15	55164A3	2	3	138	658199	0,0277778
15	50990B3	35	3	110	438378	0,0277778
15	50264B3	35	3	110	144339	0,0277778
15	55164B3	2	3	138	857316	0,0277778
15	50990B2	35	3	110	586977	0,0277778
15	55164B2	2	3	138	159097	0,0277778
15	50990A2	35	3	110	745688	0,0277778

pracy dodatkowo zmodyfikowane zgodnie z listingiem 8.2, aby wykluczyć rolę kontrolera jako przyczynę niedostępności urządzenia nadawczo-odbiorczego.

Zapytanie 8.2 pozwala z dużą dozą prawdopodobieństwa wykryć problem z transmisją (radiolinią). Uzyskano to, przez wykorzystanie funkcji agregujących *min* i *max*, w połączeniu z klauzulą *over*. Otóż, jeżeli w ramach grupy znajdują się różne kontrolery (co skutkuje różnymi identyfikatorami zwróconymi przez *min* oraz *max*), to jak wspomniano wcześniej, należy wykluczyć winę pojedynczego kontrolera. Niestety nie jest to zależność symetryczna – jeżeli w ramach skupienia występowałby wyłącznie jeden kontroler, nie można powiedzieć, że na pewno nastąpiła jego awaria, gdyż z zestawu danych nie wiadomo ile dokładnie komórek obsługuje dany kontroler. Aby stwierdzić awarię kontrolera, wszystkie sterowane przez niego urządzenia musiałyby funkcjonować nieprawidłowo. Wynik zapytania 8.2 jest dla podanego zestawu parametrów identyczny z przedstawionym w tabeli 8.1, jednakże jednoznacznie wskazuje przyczynę problemu niedostępności.

Listing 8.2: Wykrywanie problemu z transmisją.

```
select liczebnosc, cellname, obszarId, dostawcaId, kontrolerId, zdarzenieId,
round(strata, 7) as strata
from (
   select
   count(*) over (partition by start, koniec, strata) as liczebnosc,
   min(kontrolerId) over (partition by start, koniec, strata) as
   kontrolerID_min,
   max(kontrolerId) over (partition by start, koniec, strata) as
   kontrolerID_max,
   t.*
   from cell_loss t
   where t.[data] = '07-08-2010') t1
where t1.liczebnosc_s_k_str >= 10
    and t1.kontrolerID_min <> kontrolerID_max
```

W kolejnym etapie badań, zainteresowano się relacją między sektorem (który może być w uproszczeniu interpretowany jako kierunek transmisji danej komórki) a stacją bazową. Z reguły na stację bazową przypadają 3-4 sektory, więc postanowiono identyfikować te stacje, w których tylko niektóre sektory mają ograniczoną dostępność. Oznacza to, że problem nie jest związany z kontrolerem ani transmisją, co wskazuje bezpośrednio na defekt konkretnych urządzeń nadawczo-odbiorczych. Sformułowano zatem zapytanie przedstawione na listingu 8.3, którego celem jest identyfikacja wadliwych komórek.

Listing 8.3: Wykrywanie problemu z konkretnym urządzeniem nadawczo-odbiorczym.

```
select licz_s, cellname, obszarId, czestosc, czasTrwaniaH, czyWorkflow,
round(strata, 7) as strata
from (
  select
  count(*) over (partition by start, koniec, left(t.sektorId,5)) as licz_s,
  t.*
  from cell_loss t
  where t.[data] = '07-08-2010') t1
where t1.licz_s < 3 and strata between 0.3 and 0.99
order by left(sektorId,5)
```

Procedura zapytania SQL, przedstawiona na listingu 8.3, dokonuje najpierw zliczenia ile sektorów[6] obejmuje dane zdarzenie (kolumna *licz_s*) w badanym przez analityka dniu 7 czerwca 2010, a następnie wyświetla jedyne te rekordy, których liczebność jest mniejsza od trzech – jeżeli na stacji są co najmniej trzy sektory, to interesujące są tylko przypadki, w których maksymalnie dwa nie działają, a pozostały jest sprawny. Dodatkowo dodano ograniczenie na stratę (poziom niedostępności), dzięki czemu wyznaczane są zdarzenia o relatywnie wysokiej stracie (od 30%), jednakże bez uwzględnienia tych, w których komórka jest cały czas niedostępna (strata równa jeden). Sytuacje w których urządzenie nadawczo-odbiorcze jest niedostępne przez całą godzinę zwykle oznaczają jego celowe wyłączenie, dlatego nie powinny być brane pod

[6]Sektory należące do tej samej stacji bazowej różnią się wyłącznie ostatnią cyfrą, dlatego też dokonano obcięcia tego identyfikatora do pięciu miejsc, aby zliczyć wszystkie sektory.

uwagę. Oczywiście, podobnie jak poprzednio, można modyfikować okres czasu, którego dotyczy zapytanie i dowolnie zmieniać interesujący przedział straty. Wyniki opisywanego zapytania przedstawia tabela 8.2.

Tabela 8.2: Wynik zapytania 8.3

licz_s	cellname	obszarId	czestosc	czasTrwaniaH	czyWorkflow	strata
1	50269A1	36	1	6	0	0,544302
1	50419A1	11	1	2	0	0,4050926
1	50718A1	35	1	16	1	0,7516891
1	50886A3	35	1	7	0	0,7847222
1	51139A1	28	1	6	0	0,5505698
2	51493A3	20	1	4	1	0,9372989
2	51493A2	20	1	4	1	0,9372414
1	51526B2	20	2	4	1	0,7852155
2	51611A2	28	2	2	0	0,3117241
2	51611A1	28	2	2	0	0,3108929
1	54404A1	28	2	3	1	0,3474074
2	55162A1	2	2	3	0	0,5814103
2	55162A2	2	2	3	0	0,5819167
1	55162A3	2	2	2	0	0,332625
2	55322A2	28	2	2	0	0,3130769
2	55322A1	28	3	2	0	0,3128846
1	55355A1	2	3	1	1	0,4867857
1	55412A1	2	3	1	1	0,3696154
1	59132A2	42	1	1	0	0,3091257

Zapytanie 8.3 zwraca w odpowiedzi 19 rekordów, w tym identyfikatory podejrzanych urządzeń nadawczo-odbiorczych. Na podstawie wyników w tabeli 8.2, nie da się wyznaczyć ścisłej korelacji między tymi komórkami. Dlatego też należałoby je zbadać pod kątem awarii mechanicznej bądź elektrycznej, obciążenia czy panujących warunków atmosferycznych. Jednakże tego typu działania może podjąć technik z działu utrzymania sieci, mający fizyczny dostęp do urządzenia i jego warunków pracy.

Kolejną ważną kwestią jest wykrywanie urządzeń, którymi należy zająć się natychmiast (np. poddać oględzinom i ewentualnym naprawom). Po konsultacjach z ekspertem zaproponowano, by były to komórki niedostępne przez dłuższy okres czasu (np. więcej niż 5 dni) oraz z określonym poziomem straty (np. 30%). W tym celu skonstruowano zapytanie przedstawione na listingu 8.4. Wynik zapytania zawarto w postaci tabeli 8.3.

Listing 8.4: Urządzenia, które pilnie należy poddać oględzinom.

```
select distinct cellname
from cell_loss
where koniec - [start] > 5
      and strata > 0.3
```

Tabela 8.3: Wynik zapytania 8.4

cellname
53774D1
58331A1
59126A1
58331A3
58331A2
59166A1
50291A2
50010A1

Zgodnie z danymi zawartymi w tabeli 8.3, wykryto aż osiem komórek, które przynajmniej raz miały problem z dostępnością przez ponad pięć dni z rzędu. Jest to oczywiście sytuacja bardzo niekorzystna z punktu widzenia ciągłości i jakości oferowanych usług. Powinno się niezwłocznie poddać ewaluacji komórki z tej listy, biorąc pod uwagę rzeczywiste warunki ich pracy i dokonać ich przestawiania, naprawy bądź nawet wymiany. Są to niewątpliwie najsłabsze ogniwa całej sieci. Tego typu mechanizm identyfikacji problematycznych urządzeń, powinien zdaniem autora, zostać dołączony do automatycznego systemu monitorującego ich pracę.

Eksperyment 3. Wyznaczenie optymalnych parametrów sterujących dla gęstościowych algorytmów grupowania, na podstawie zbioru cell_loss

Kolejnym problemem omawianym w ramach niniejszej pracy, jest dobór parametrów startowych dla algorytmów gęstościowych. Wydawałoby się, że zgodnie z informacjami teoretycznymi zawartymi w rozdziale 5, najlepiej jest w każdym możliwym przypadku stosować algorytm OPTICS, ponieważ wymaga on określenia jednego parametru (minimalnej liczby obiektów w grupie *MinPts*) zamiast dwóch, jak to ma miejsce dla metody DBSCAN. Jednakże, jak stwierdzono podczas przeprowadzania eksperymentów obliczeniowych na potrzeby niniejszej rozprawy, czas działania algorytmu OPTICS, bywa wielokrotnie dłuższy od czasu potrzebnemu algorytmowi DBSCAN do wygenerowania kilku grupowań. Autorzy algorytmu OPTICS podają w [4], że czas działania OPTICS jest o 1,6 raza dłuższy, niż w przypadku metody DBSCAN. Jednakże brak jest precyzyjnych informacji na temat tego, jak został wyznaczony wspomniany współczynnik. Grupowanie algorytmem OPTICS, tabeli *interface_aggreg* (ograniczonej do nieco ponad 40 tysięcy obiektów i 6 atrybutów[7]), wchodzącej w skład zbioru danych złożonych *ap_loss*, zostało przerwane po 30 godzinach oczekiwania na wynik[8]. Ten sam zbiór poddany działaniu algorytmu DBSCAN (dla określonego *Eps* i *MinPts*), udało się podzielić na skupienia, korzystając z systemu *DensGroup*, w czasie rzędu 3-4 godzin. Nawet zakładając, że OPTICS umożliwia szybkie wygenerowanie sześciu podziałów (bo tyle atrybutów jest branych

[7]Motywacja do wprowadzenia takich ograniczeń, została wyjaśniona szczegółowo w sekcji poświęconej analizie zbioru ap_loss.

[8]Eksperyment powtórzono również na innym komputerze oraz korzystając z programu Weka (omówionego w rozdziale 3). Jedyną zmianą dokonaną przez autora rozprawy w oprogramowaniu Weka, było zaimplementowanie funkcji podobieństwa obiektów CommonDataObject, tak by była ona zgodna z definicją przyjętą w autorskim oprogramowaniu DensGroup – liczba cech różnych została przyjęta jako funkcja podobieństwa.

pod uwagę w fazie grupowania) dla stałego *MinPts*, to wykonanie tego samego zadania algorytmem DBSCAN trwałoby maksimum 24 godziny. Pozwala to założyć, że współczynnik 1,6 podany przez autorów algorytmu, dotyczył tylko ograniczonej liczby zbiorów przez nich testowanych i może to być wartość przybliżona, nieadekwatna dla bardziej złożonych przypadków. Biorąc pod uwagę przedstawione argumenty, autor rozprawy sugeruje by używać algorytmu OPTICS, przede wszystkim dla zbiorów o dużej wymiarowości (liczbie atrybutów) i relatywnie małej liczbie obiektów. Przykładowo, jeżeli przetwarzane są mikromacierze (wykorzystywane przy badaniu ekspresji genów), liczące setki bądź tysiące atrybutów, możliwość szybkiego wygenerowania setek różnych podziałów na skupienia, z całą pewnością będzie zaletą[9]. Natomiast algorytm DBSCAN powinien być stosowany w pozostałych przypadkach, a szczególnie tam, gdzie liczy się czas działania algorytmu (kosztem np. jakości takiego podziału) – często w środowiskach biznesowych niedopuszczalne jest oczekiwanie powyżej 24 godzin, ze względu na konieczność adaptacji, do zmieniających się z dnia na dzień, oczekiwań klientów i wymagań rynku. Dlatego też w systemie *DensGroup* zaimplementowano oba algorytmy i to do użytkownika należy wybór właściwego w danym momencie.

Autor rozprawy proponuje następującą heurystykę do wyznaczania optymalnych parametrów algorytmów gęstościowych (przy przetwarzaniu danych złożonych). Utworzona liczba skupień powinna stanowić ok 6% liczności danego zbioru, liczba obiektów wchodzących w skład trzech najliczniejszych grup powinna być większa niż 2% całkowitej, a liczba obiektów izolowanych (nie przypisanych do żadnego skupienia) powinna być zredukowana do minimum. Większa liczba obiektów w grupach, bądź występowanie sporej ilości szumu informacyjnego mogłoby znacząco utrudnić analizę i interpretację utworzonej struktury, ponieważ do grupy obiektów izolowanych klasyfikowane są potencjalnie niezwiązane ze sobą instancje. Taka metodyka postępowania powinna skutkować utworzeniem grup, które mogą zostać przeanalizowane w rozsądnym czasie i posiadają w miarę równomierny rozkład. Ponadto, nie wymaga ona dużych nakładów obliczeniowych (poza kosztem związanym z generowaniem podziału na skupienia). Dlatego też w następnym eksperymencie zostanie wygenerowanych kilka podziałów na grupy, dla różnych wartości parametrów startowych *Eps* i *MinPts*. Dla każdego podziału określono następujące wielkości: liczbę stworzonych grup, liczbę obiektów zaklasyfikowanych jako wartości izolowane (szum informacyjny) oraz liczbę obiektów w trzech najliczniejszych grupach. Podczas grupowania brane są pod uwagę wszystkie atrybuty, za wyjątkiem identyfikatora komórki (*cellname*), identyfikatora zdarzenia (*zdarzenieId*), godziny (i daty) rozpoczęcia (*start*) jak również końca pomiarów (*koniec*) oraz dokładnej niedostępności komórki w danej godzinie (*strata*). Powodem wykluczenia tych atrybutów była zbyt duża liczba unikalnych wartości, w stosunku do liczby obiektów w zbiorze oraz sugestie eksperta dziedzinowego. Wymienione atrybuty wpływałyby niekorzystnie na spójność grup i sztucznie powiększały ich separacje. Wyniki opisywanego eksperymentu zostały zamieszone w tabeli 8.4.

Wyniki zilustrowane za pomocą tabeli 8.4 jednoznacznie sugerują, że już bardzo niewielka zmiana wartości pierwszego parametru sterującego *Eps*, ma znaczący wpływ na liczbę utworzo-

[9]Po wygenerowaniu uporządkowania przez algorytm OPTICS, wystarczy na podstawie wartości odległości wewnętrznej i osiągalnej, odpowiednio podzielić je na grupy. Wymaga to jednokrotnego przeglądu takiego uporządkowania. Szczegóły na temat generowania struktury skupień, na podstawie danych generowanych przez OPTICS, znajdują się w rozdziale 5.

Tabela 8.4: Wyznaczanie optymalnych parametrów algorytmu gęstościowego dla zbioru *cell_loss*

Eps	**MinPts**	**Liczba grup**	**Wartości izolowane**	**Pierwsza największa grupa**	**Druga największa grupa**	**Trzecia największa grupa**
0	1	107265	Brak	13	10	8
0	2	30236	77029	13	10	8
0	3	3411	130679	13	10	8
1	1	7934	Brak	6279	5722	5653
1	2	5683	2251	6279	5722	5653
1	3	4630	4357	6279	5722	5653
2	1	1869	Brak	89380	19285	5468
2	2	1434	435	89380	19285	5468
2	3	1196	911	89380	19285	5468
3	1	136	Brak	142901	36	18
3	2	103	33	142901	36	18
3	3	88	63	142901	36	18
4	1	3	Brak	143478	7	1
4	2	2	1	143478	7	Brak
4	3	2	1	143478	7	Brak
5	1	1	Brak	143478	Brak	Brak
5	2	1	Brak	143486	Brak	Brak
5	3	1	Brak	143486	Brak	Brak

nych grup. Największą zmianę w tym zakresie, można zaobserwować przy zwiększaniu wartości tego parametru z zero na jeden. W literaturze przedmiotu [16] często spotyka się podejście wyboru tych parametrów sterujących pracą gęstościowego algorytmu grupowania, dla których następuje gwałtowna zmiana w liczbie grup, wartości założonego wskaźnika oceny jakości[10], lub w ogólnie pojętym przydziale obiektów do skupień.

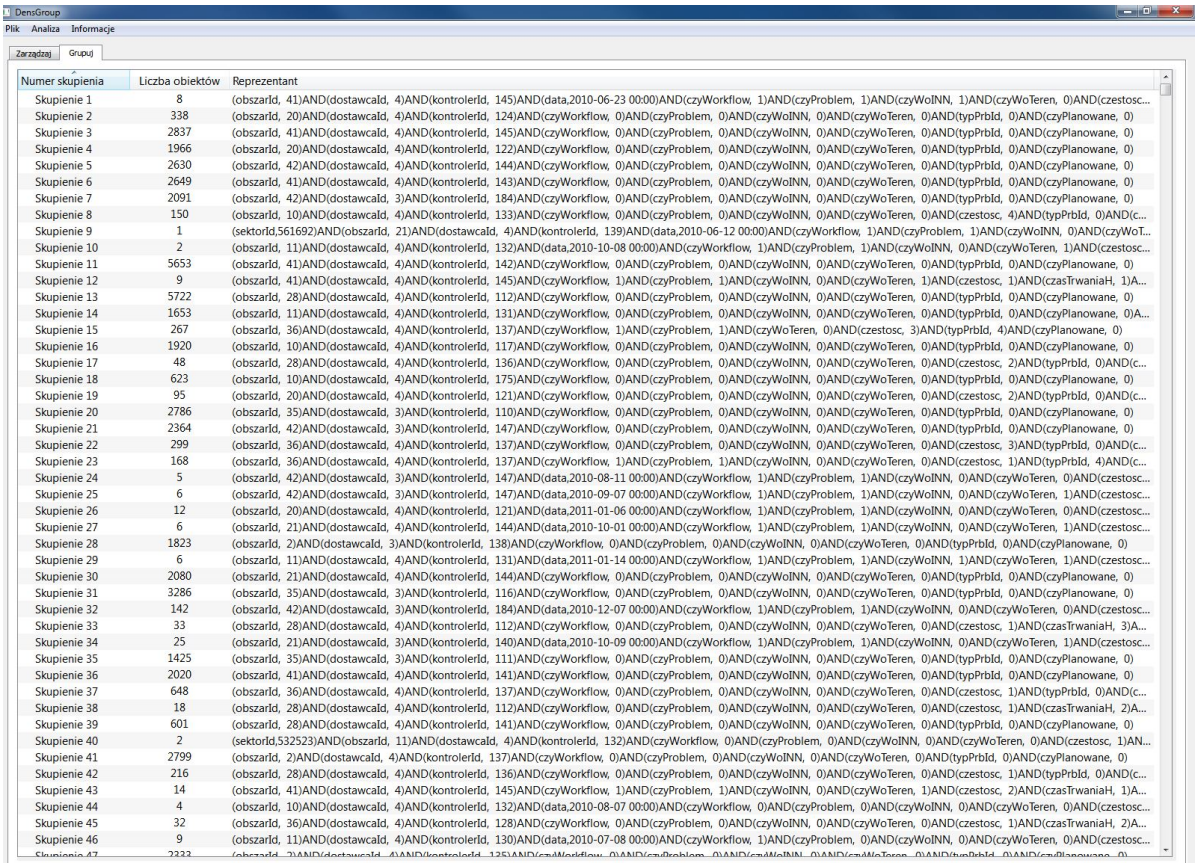
Drugi parametr wejściowy (*MinPts*) ma natomiast duży wpływ na rozmiar grupy wartości izolowanych (powstawanie szumu informacyjnego). Generalnie im jest on większy, tym więcej obiektów ze zbioru danych, jest klasyfikowanych jako szum informacyjny. Nie narusza on jednak rozkładu obiektów do najliczniejszych grup — zmiany wartości parametru *MinPts* nie mają żadnego wpływu na wielkość tych skupień (za wyjątkiem sytuacji, w której promień sąsiedztwa jest ustawiony na wartość pięć i pojedynczy obiekt tworzy skupienie). Interesujący jest również fakt, że pewne obiekty zostają zaklasyfikowane jako szum informacyjny, wyłącznie jeśli parametr *MinPts* jest ustawiony na wartość 2 lub wyższą. Może to świadczyć o niskiej spójności

[10]Autor już podczas realizacji pracy magisterskiej [71] napisał rozdział odnośnie zagadnienia jakości grupowania, w którym omówiono najpopularniejsze metody oceny, jednakże na potrzeby niniejszej rozprawy nie zdecydowano się na uwzględnienie którejkolwiek z nich. Nadrzędnym celem rozprawy jest bowiem wydobywanie wiedzy (np. na podstawie złożonej struktury skupień), zatem nawet w sytuacji w której utworzone zostałyby przykładowo tylko dwie grupy (jedna agregująca 99% wszystkich obiektów oraz druga bardzo mała), a miara oceny odrzucałaby taki podział (ze względu na niską spójność), wśród elementów mniejszego skupienia można byłoby wyznaczyć obiekty odstające parametrami swojej pracy od innych co dostarczyłoby interesujących informacji. Przykład podobnej sytuacji został zademonstrowany dla eksperymentu 8 i tabeli *interface_aggreg*.

grup, tzn. obiekty w ramach skupień przypuszczalnie charakteryzują się niską gęstością. Zatem korzystając ze wcześniej przedstawionego sposobu doboru parametrów początkowych, zostały ustalone następujące wartości: **Eps = 1**, **MinPts = 1**. Takie też będą używane podczas przeprowadzania kolejnego eksperymentu.

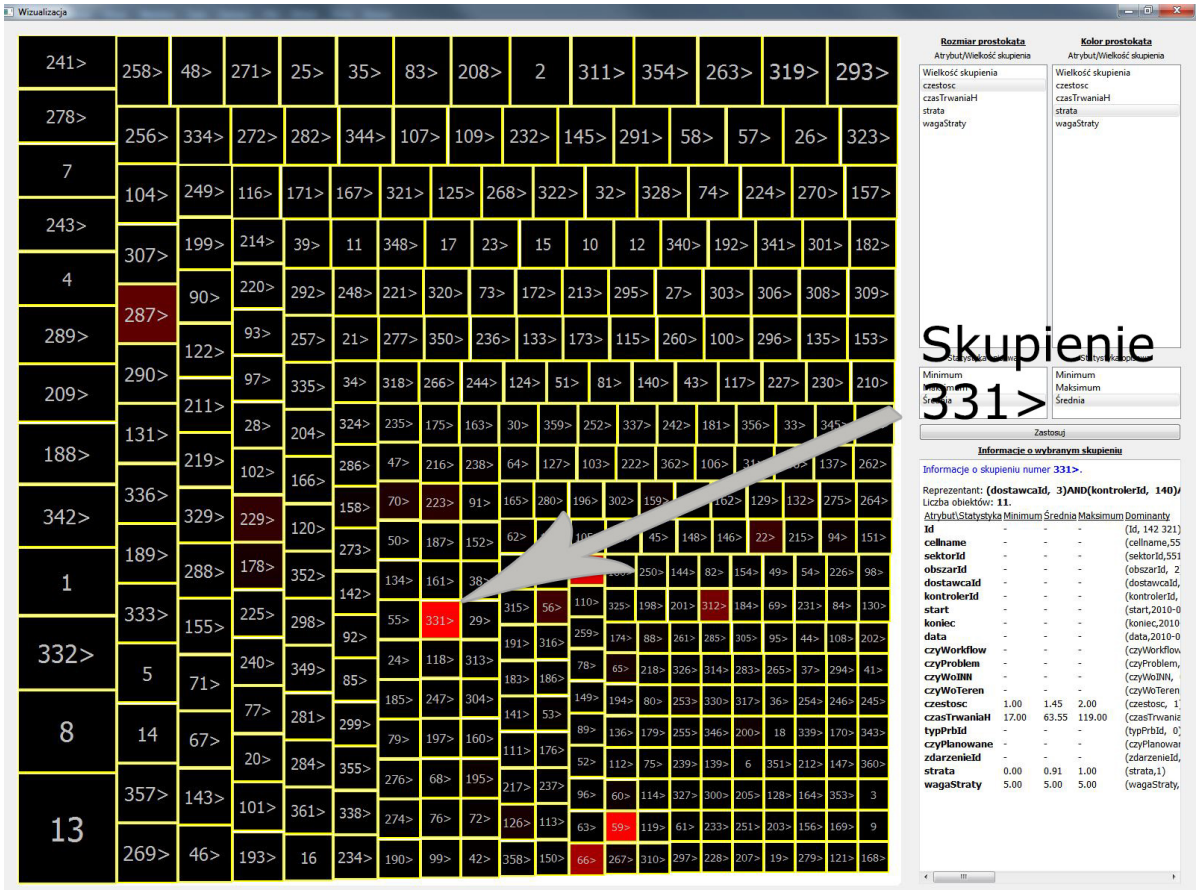
Eksperyment 4. Analiza przydatności zastosowania systemu *DensGroup* w procesie odkrywania wiedzy, na podstawie zbioru cell_loss

Ostatni eksperyment przeprowadzony dla zbioru *cell_loss*, dotyczył zastosowania możliwości autorskiego systemu *DensGroup*, celem wydobycia użytecznej wiedzy. Dokonano zatem grupowania algorytmem DBSCAN, dla ustalonych wartości parametrów *Eps = 1*, *MinPts = 1* uzyskując niemalże osiem tysięcy grup. Jak umotywowano w rozdziale 7, jest to zbyt duża liczba skupień do analizy w rozsądnym czasie. Dlatego też kolejny krok polegał na zastosowaniu algorytmu AHC, by uzyskać dwupoziomową strukturę hierarchiczną, a tym samym znacząco zredukować wynikową liczbę skupień. Skupienia te poddawane były wówczas wizualizacji, za pomocą opisywanej w rozdziale 6, metody map prostokątów. Wynik generowania grup, przy użyciu algorytmu DBSCAN, prezentuje rysunek 8.4.



Rysunek 8.4: Wynik zastosowania algorytmu DBSCAN na zbiorze *cell_loss*.
Źródło: Opracowanie własne

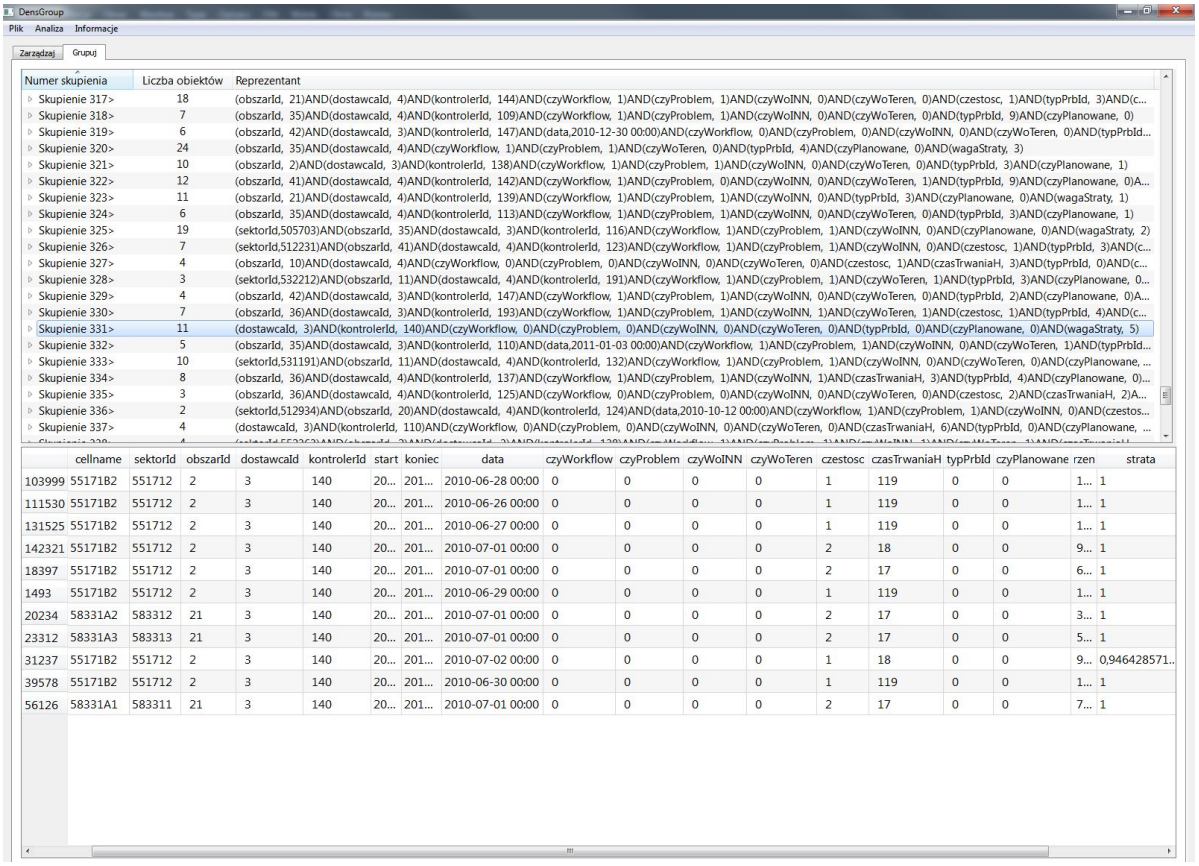
Zgodnie z oczekiwaniami, algorytm analizy skupień wygenerował 7934 grupy o liczności od jednego do 6279 obiektów. Mimo intuicyjnej metody tworzenia reprezentantów skupień, wykorzystującej koniunkcję logiczną deskryptorów, struktura na rysunku 8.4 jest bardzo trudna w analizie, zatem zdecydowano się wykorzystać algorytm aglomeracyjny jako metodę agregacji skupień. Potencjalnie pozwoli to na uproszczenie uzyskanej struktury i łatwiejszą interpretację wygenerowanych powiązań między obiektami. Ustalono liczbę skupień (generowanych przez metodę AHC) na podstawie progu podobieństwa wynoszącego pięć[11]. Oznacza to, że podczas generowania struktury hierarchicznej, jeżeli na danym poziomie dwa obiekty między sobą różnią się więcej niż pięcioma cechami, zapamiętywany jest stan hierarchii i zostaje ona przycięta do tego poziomu. Jako kryterium łączenia skupień, zastosowano metodę średniego wiązania. W wyniku działania algorytmu AHC wygenerowano 368 skupień. Jest to jednak w dalszym ciągu zbyt duża liczba do skutecznej analizy (bazując wyłącznie na opisie reprezentanta grupy), dlatego też zdecydowano się skorzystać z zaimplementowanej w systemie *DensGroup* metody wizualizacji.



Rysunek 8.5: Wizualizacja struktury grup, utworzonej dla zbioru *cell_loss*.
Źródło: Opracowanie własne

[11]Wartość pięć została dobrana arbitralnie jako kompromis między najkrótszym a najdłuższym reprezentantem.

Biorąc pod uwagę strukturę zbioru *cell_loss*, w wizualizacji przedstawiającej rozkład grup, skupiono się na parametrach częstości występowania zdarzeń (w ciągu danego dnia pomiarowego) oraz procencie niedostępności urządzenia. Dlatego też w wizualizacji (przedstawionej na rysunku 8.5), rozmiar prostokąta symbolizuje średnią częstość zdarzeń, natomiast kolor określa poziom straty (w ramach skupień). Im prostokąt posiada większe pole, tym częściej notowane były zdarzenia (np. niedostępność urządzenia, wynikająca z braku zasilania) związane z pracą urządzeń w danej grupie. Natomiast im bardziej jasno-czerwony kolor danego prostokąta, tym większą stratę (niedostępność) zarejestrowano. Priorytetem dla operatora sieci jest oczywiście dostępność usług, zatem to ten drugi parametr jest w tej analizie ważniejszy. Na tej podstawie udało się zlokalizować skupienie numer *331>*. Badając informacje o danym skupieniu (przedstawione w prawym dolnym rogu rysunku 8.5), widać że skupienie to liczy 11 obiektów (opisujących pracę trzech komórek), których średnia strata wynosi aż 91%. Dodatkowo, maksymalnie dwa razy zostały w ciągu danego dnia zarejestrowane zdarzenia, związane z pracą tych komórek. Kolejnym krokiem jest zatem analiza wnętrza tego skupienia, pod kątem parametrów pracy konkretnych urządzeń sieciowych, wchodzących w jego skład. Zostało to zilustrowane na rysunku 8.6.



Rysunek 8.6: Zawartość skupienia numer *331>*.
Źródło: Opracowanie własne

Zawartość interesującego skupienia (przedstawiona na rysunku 8.6) ujawnia, że problema-

tyczne są cztery urządzenia o identyfikatorach *55171B2*, *58331A2*, *58331A3*, *58331A1*. Wszystkie wymienione komórki pochodzą od tego samego dostawcy i są sterowane przez kontroler 140. Trzy spośród wymienionych urządzeń (*58331A3*, *58331A2*, *58331A1*), pracują w pobliskich sektorach. Wszystkie zarejestrowane zdarzenia miały miejsce między 26-06-2010 a 2-07-2010, co można uznać za początek okresu wakacyjnego. Żadne z tych zdarzeń nie było planowane i nie było zleceń naprawczych w przedstawionym okresie czasu. Pod kątem czasu trwania niechlubnie wyróżnia się urządzenie *55171B2*, ponieważ było niedostępne aż 119 godzin. Dodatkowo jednokrotnie miało ono stratę poniżej 100%, co wykluczałoby celowe wyłączenie tego urządzenia (jak mogłyby nieustanne 100% straty sugerować). Udało się zatem zidentyfikować co najmniej jedno problematyczne urządzenie (potencjalnie cztery), którym należałoby się zająć w pierwszej kolejności i sprawdzić jego działanie na miejscu. Co istotne, urządzenia *58331A3*, *58331A2*, *58331A1* zostały już wcześniej zidentyfikowane za pomocą zapytania SQL, przedstawionego na listingu 8.4, jednakże żadna z zaprezentowanych metod nie wykryła urządzenia *55171B2*. Potwierdza to użyteczność systemu *DensGroup* i algorytmów analizy skupień, w odkrywaniu wiedzy z danych złożonych.

8.2 Wydobywanie wiedzy ze zbioru *ap_loss*

Podobnie jak dla poprzedniego omawianego zbioru, pierwszy przeprowadzony eksperyment dotyczy wykorzystania graficznych metod opisu danych, w celu zapoznania się z ich strukturą i potencjalnie odkryciu użytecznej wiedzy. Jak zaznaczono w rozdziale 2 zbiór *ap_loss* składa się ostatecznie z czterech tabel, które ze względu na liczność agregowanych obiektów oraz brakujące wartości[12], zostaną poddane analizie osobno.

Eksperyment 5. Analiza przydatności wykorzystania histogramów, jako metody opisu danych, w procesie odkrywania wiedzy dla zbioru ap_loss

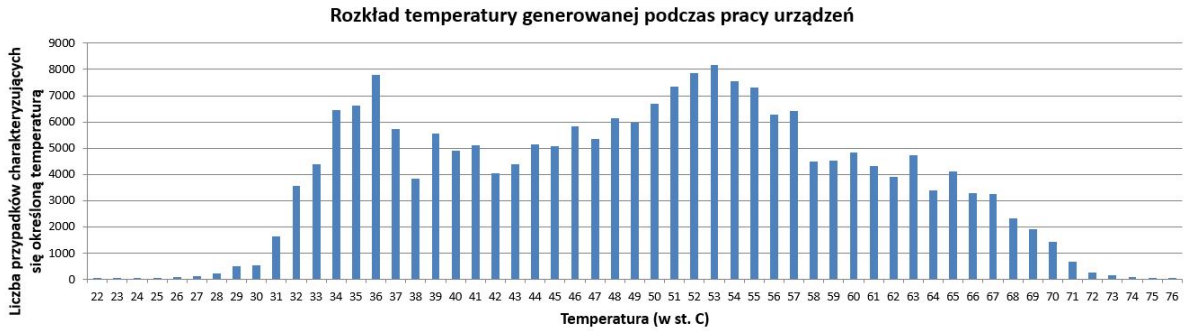
Wykresy prezentujące rozkład unikalnych wartości przyjmowanych przez poszczególne atrybuty tabeli *devices*, nie dostarczyły nowych informacji na temat analizowanego zbioru. Urządzenia do systemu monitorującego dodawane były zazwyczaj pojedynczo, w różnych odstępach czasu, a co najmniej dziewięć z nich to tzw. punkty dostępu (ang. *access point*), czyli najczęściej spotykane urządzenia zapewniające klientom (np. poszczególnym komputerom) bezprzewodowy dostęp do sieci, co można wywnioskować po przyrostku *AP* w nazwie oraz zgodności ich typu (kolumna *device_type_id*).

W przypadku tabeli *device_aggreg* sytuacja była nieco inna. Analizując wykres przedstawiający rozkład wykorzystania mocy procesora[13] (biorąc pod uwagę wszystkie urządzenia), zilustrowany na rysunku 8.7, można stwierdzić, że żaden element sieci nie jest nadmiernie obciążony. Średnie wykorzystanie (mocy) procesora dla wszystkich urządzeń wynosi 37%, jednakże blisko 21 tysięcy wpisów charakteryzuje się obciążeniem na poziomie zaledwie 28%. Podobnie wykorzystanie pamięci było niskie, ponieważ w bazie zarejestrowano wartości w przedziale od

[12] Niektóre tabele nie posiadają informacji o pracy poszczególnych urządzeń, ponieważ pochodzą bezpośrednio od klientów co nie gwarantuje integralności danych.

[13] Dla poprawy czytelności wykresu zastosowano skalę logarytmiczną na osi rzędnych.

Rysunek 8.7: Rozkład wykorzystania mocy procesora przez wszystkie urządzenia.
Źródło: Opracowanie własne



Rysunek 8.8: Rozkład temperatury generowanej podczas pracy urządzeń.
Źródło: Opracowanie własne

21 do 30 procent. Średnie wykorzystanie dostępnej pamięci wynosiło zaledwie 23%, co potwierdza tezę o optymalnym doborze możliwości sprzętowych do ich przewidywanego obciążenia.

Interesujący w tym kontekście jest rozkład zarejestrowanych temperatur, przedstawiony na rysunku 8.8. Wynika z niego bezpośrednio, że zdarzają się przypadki, w których temperatura przekracza 70 stopni. Jest to nadal dopuszczalna wartość, ponieważ przyjmuje się, że bezpieczna temperatura pracy dla najczęściej spotykanych urządzeń sieciowych, to temperatura poniżej stu stopni, jednakże należałoby zbadać, co wpływa na najwyższe zarejestrowane wartości temperatur i czy można tę sytuację poprawić. Brak jest niestety informacji o warunkach zewnętrznych, panujących podczas pracy urządzenia, jak również o dokładnej strukturze sieci, zatem jedynym czynnikiem, który można bezpośrednio zbadać korzystając z posiadanych informacji, jest ewentualna korelacja między obciążeniem procesora a wysokimi temperaturami pracy. Potencjalnie należy się spodziewać, że im wyższa zarejestrowana temperatura, tym wyższe obciążenie procesora danego urządzenia. Zostanie to zweryfikowane przez stworzenie odpowiedniego zapytania SQL, opisanego w dalszej części niniejszego rozdziału.

W tabeli *interfejs_aggreg* zawarte są informacje na temat ruchu generowanego przez poszczególne urządzenia, z podziałem na interfejsy. Korelacja urządzeń z przynależącymi do niego interfejsami potwierdziła jedynie ich równomierne obciążenie – żadne z urządzeń nie wyróżnia się, pod kątem przesyłanej lub odbieranej liczby pakietów czy bajtów. Jednakże we wspo-

mnianej tabeli występują również dwa bardzo interesujące (z punktu widzenia analizy danych) atrybuty: *rx_err_rate* oraz *tx_err_rate* oznaczające kolejno liczbę błędów odbioru i liczbę błędów transmisji, podczas wymiany informacji (połączenia). Rozkład częstości występowania poszczególnych wartości przyjmowanych przez *tx_err_rate*, przedstawiono w tabeli 8.5.

Tabela 8.5: Rozkład liczby błędów transmisji

Wartość	Liczba wystąpień
0	406793
1	1
2	1
3	1
5	1
7	1
10	1

Z tabeli 8.5 wynika, że biorąc pod uwagę wszystkie urządzenia oraz próby pomiarowe, błędy transmisji występują bardzo rzadko (jednokrotnie zarejestrowano od jednego do dziesięciu błędnie przesłanych pakietów). Zważywszy na fakt, że w zbiorze danych zanotowano transmisje liczące do 744120 pakietów (średnio ponad 40 tysięcy), wartości błędów są pomijalne. Niemniej postanowiono sprawdzić, czy istnieje korelacja między jakimiś urządzeniami a błędami przy transmisji (co zostało zrealizowane przez wykonanie zapytania SQL).

Rysunek 8.9: Liczba błędów odbioru, zarejestrowana dla wszystkich urządzeń.
Źródło: Opracowanie własne

Podobną analizę przeprowadzono dla parametru *rx_err_rate*, czyli liczby błędów odbioru. Ponieważ unikalnych wartości przyjmowanych przez ten atrybut było znacznie więcej niż w poprzednim przypadku, w tym celu stworzono histogram zaprezentowany na rysunku 8.9. Na podstawie wykresu 8.9 można stwierdzić, że błędy odbioru pojawiają się zdecydowanie częściej i mogą mieć realny wpływ na jakość całego połączenia. Dlatego też koniecznym wydaje się zestawienie liczby błędów, generowanych przez poszczególne urządzenia (co również najprościej wykonać, tworząc zapytanie do relacyjnej bazy).

W przypadku ostatniej analizowanej tabeli czyli *stations_aggreg*, która przechowuje statystyki związane z pracą urządzeń klienckich podłączonych do stacji bazowej, najbardziej istotne atrybuty (pod kątem eksploracyjnej analizy danych) to *rssi* – wskaźnik mocy odbieranego sygnału radiowego, *tx_power* – moc wyjściowa nadajnika oraz *device_id* – identyfikator podłączonego urządzenia. Niestety pierwszy parametr (*rssi*) zmienia się wraz ze zmianą położenia stacji klienckiej bądź nawet warunków zewnętrznych, przez co nie można na jego podstawie stwierdzić, że odbierany poziom sygnału jest niski, więc należy zmienić lokalizację stacji. Moc wyjściowa nadajnika (wyrażana w decybelach) jest stała dla danego urządzenia (stacji) i nie da się wprost stwierdzić, że zwiększenie tego parametru wpłynie pozytywnie na zasięg czy przepustowość transmisji. Zasięg łącza bezprzewodowego zależny jest bowiem również od zysku energetycznego anteny[14] oraz czułości odbiornika[15]. Przykładowo, zastosowanie anteny o większym zysku (16 dBi) w połączeniu z nadajnikiem o mniejszej mocy (5 dBm), zapewnia na dystansie 300 metrów taki sam poziom sygnału, jak antena o zysku 6 dBi w połączeniu z nadajnikiem o większej mocy (15 dBi) na dystansie 100 metrów [13]. Często spotykane są zatem nadajniki z ustawioną małą mocą wyjściową, skorelowane z antenami o dużym zysku energetycznym. Dlatego też dalszej analizie zostanie poddany atrybut *device_id*. Wykres 8.10 przedstawia częstość występowania poszczególnych urządzeń w statystykach.

Rysunek 8.10: Częstość występowania poszczególnych urządzeń w tabeli *stations_aggreg*.
Źródło: Opracowanie własne

Z wykresu 8.10 wyraźnie wynika, że urządzenie którego tyczy się najwięcej pomiarów, posiada identyfikator 54. Dotyczy go ponad 18 tysięcy rekordów, co stanowi niemalże 20% całego zbioru. Wydawałoby się, że duża częstość występowania świadczy o tym, iż jest to jedno z kluczowych urządzeń w całej sieci lub jest nadmiernie obciążone. Jednakże po bliższej analizie okazuje się, że jest to jedyne urządzenie, dla którego statystyki były zapisywane od końca marca do września 2010 roku. Pozostałe urządzenia monitorowane były znacznie krócej (np. miesiąc lub dwa). Przedstawione argumenty niestety uniemożliwiają sensowną analizę tabeli *stations_aggreg*, mimo że nawet ekspert dziedzinowy początkowo zakładał inny scenariusz.

[14]Zysk energetyczny określa stosunek mocy sygnału emitowanego przez daną antenę, do mocy sygnału emitowanego przez antenę wymodelowaną matematycznie.

[15]Czułość odbiornika, rozumiana jest jako poziom sygnału potrzebny do poprawnego odebrania danych, przy określonej prędkości transmisji.

Eksperyment 6. Analiza możliwości zastosowania klauzul grupujących i funkcji agregujących języka SQL, celem znalezienia korelacji między obiektami (bądź atrybutami) zbioru ap_loss

Kolejne eksperymenty polegały na utworzeniu poleceń SQL, które mają na celu zweryfikowanie korelacji, wyszczególnionych podczas analiz z wykorzystaniem histogramów. Pierwsze zapytanie, zapisane tu jako listing 8.5, polegało na wyborze tych rekordów (z tabeli *device_aggreg*), dla których zarejestrowana temperatura jest większa niż 70 stopni, pogrupowaniu ich według poszczególnych urządzeń oraz wyznaczeniu, w ramach utworzonych grup, średniej temperatury jak również minimalnego, średniego i maksymalnego obciążenia procesora. Wynik tego polecenia SQL przedstawia tabela 8.6.

Listing 8.5: Korelacja między temperaturą a obciążeniem procesora danych urządzeń.

```
SELECT name as nazwa, avg(convert(int, temps)) as srednia_temp,
  avg(convert(int, cpu_usages)) as srednie_cpu,
  max(convert(int, cpu_usages)) as max_cpu,
  min(convert(int, cpu_usages)) as min_cpu,
FROM device_aggreg da, devices d
WHERE da.device_id = d.id AND temps > 70
GROUP BY name
ORDER BY name
```

Tabela 8.6: Wynik zapytania 8.5

nazwa	srednia_temp	srednie_cpu	max_cpu	min_cpu
AP-Carriage-House-1	71	29	45	26
AP-Hilton-1	72	46	55	40
AP-Luxor-1	72	49	66	39
AP-Luxor-2	71	33	53	28
AP-Plaza-1	71	34	43	28
AP-Primm-1	71	49	59	41

Rezultat przedstawiony w tabeli 8.6 wskazuje jednoznacznie, że wszystkie urządzenia posiadają dość wysoką temperaturę biorąc pod uwagę, iż średnie obciążenie procesora w każdym przypadku było poniżej 50%. Szczególnie niekorzystnie wyróżniają się urządzenia *AP-Carriage-House-1*, *AP-Luxor-2*, *AP-Plaza-1*, które to mimo wykorzystania procesora zaledwie w około trzydziestu procentach, charakteryzują się średnią zarejestrowaną temperaturą 71% Celsjusza. Naturalnie należałoby powiązać te wyniki z warunkami atmosferycznymi, panującymi w miejscu pracy urządzenia w momencie zapisywania statystyk, ale nawet jeżeli zewnętrzna temperatura okazałaby się wysoka, należałoby pomyśleć o zapewnieniu dodatkowego chłodzenia dla wyszczególnionych urządzeń. Powinno to korzystnie wpłynąć na jakość, a przede wszystkim ich czas pracy.

Kolejne utworzone zapytanie SQL miało za zadanie wykryć, które urządzenia generują błędy podczas transmisji i jaka jest wówczas średnia wartość tego parametru. Dokonano zatem grupowania danych z tabeli *inteface_aggreg* na podstawie nazwy urządzenia oraz dla

Listing 8.6: Urządzenia wyróżniające się pod kątem liczby błedów transmisji.

```
SELECT *
FROM (
  SELECT name as nazwa,
  round( avg(convert(float, tx_err_rate)), 7) as sredni_blad_tx
  FROM devices d, interface_aggreg ia
  WHERE d.id = ia.device_id
  GROUP BY name
) t
WHERE sredni_blad_tx > 0
```

każdej z grup wyznaczono średnią z wartości przyjmowanych przez atrybut *tx_err_rate*. Następnie wybrano jedynie te urządzenia, które generowały jakikolwiek odsetek błędów podczas transmisji. Strukturę zapytania przedstawiono na listingu 8.6, natomiast jego rezultat zawarto w postaci tabeli 8.7.

Tabela 8.7: Wynik zapytania 8.6

nazwa	**sredni_blad_tx**
AP-Plaza-1	0,0007737

Rezultat zapytania 8.6 stanowi tylko jeden rekord. Oznacza to, że tylko punkt dostępowy o nazwie *AP-Plaza-1*, generuje jakiekolwiek błędy transmisji. Oczywiście liczba błędów ogółem jest bardzo niska (jak zapisano w tabeli 8.5), co implikuje równie niską wyznaczoną średnią. Jednakże udało się wykryć pojedyncze źródło błędów, co wskazuje, że urządzenie *AP-Plaza-1* może pracować w trudnych warunkach lub jest nieoptymalnie rozmieszczone. Fakt niewłaściwej lokalizacji urządzenia został ostatecznie potwierdzony przez eksperta dziedzinowego.

Listing 8.7: Urządzenia wyróżniające się pod kątem liczby błędów odbioru.

```
SELECT *
FROM
(
  SELECT name as nazwa,
  round( avg(convert(float, tx_err_rate)), 7) as sredni_blad_rx
  FROM devices d, interface_aggreg ia
  WHERE d.id = ia.device_id
  GROUP BY name
) t
WHERE sredni_blad_rx > 0
ORDER BY sredni_blad_rx DESC
```

Podobną, do przeprowadzonej powyżej, analizę wykonano dla parametru *rx_err_rate*, symbolizującego liczbę błędów podczas odbioru danych. Stworzono zatem analogiczne zapytanie SQL, zaprezentowane na listingu 8.7. Wynik zapytania 8.7 przedstawia tabela 8.8. Tym razem zostało zwróconych aż sześć urządzeń, które w różnym stopniu wykazują błędy odczytu. Jednakże ponownie spośród tych sześciu urządzeń, najbardziej problematyczny jest punkt dostępu

Tabela 8.8: Wynik zapytania 8.7

nazwa	sredni_blad_rx
AP-Plaza-1	0,6509173
AP-Hilton-1	0,1099883
AP-Primm-1	0,0060556
AP-Luxor-2	0,0011591
AP-Plaza-2	0,0007173
AP-DonSmith-1	0,0005735

AP-Plaza-1. Potwierdza to dodatkowo spostrzeżenie o niewłaściwym umiejscowieniu urządzenia. Podobną tezę należy postawić również w przypadku punktu dostępowego *AP-Hilton-1*. Średnie błędy odbioru dla pozostałych urządzeń są już o dwa rzędy wielkości niższe, przez co bardziej należy wskazywać na różnice w jakości kart sieciowych urządzeń (komputerów) klienckich, niż rozmieszczenie czy wady samych punktów dostępowych. W laptopach (które są najczęściej podłączanymi urządzeniami do punktów dostępowych) stosowanych jest kilka popularnych układów odpowiedzialnych za transmisję bezprzewodową oraz stosuje się różne rodzaje obudowy i anten, które mają bezpośredni wpływ na tłumienie sygnału, a ostatecznie na zasięg i jakość połączenia. Niemniej wskazany jest audyt wyszczególnionych urządzeń.

Listing 8.8: Potencjalnie awaryjne urządzenia.

```
SELECT DISTINCT * FROM
(
  SELECT
      count(*) over (partition by name) as czestosc,
      d.name as nazwa,
      d.created_on as utworzono
  FROM device_aggreg da, devices d
  WHERE uptime < 900
  AND d.id = da.device_id
) t
ORDER BY czestosc DESC
```

Ostatnie utworzone zapytanie SQL (zaprezentowane na listingu 8.8) miało na celu wykrycie potencjalnie często resetujących się urządzeń. Wykorzystano w nim m.in. atrybut *uptime* z tabeli *device_aggreg*, symbolizujący czas nieprzerwanej pracy urządzenia (wyrażony w sekundach). Idea tego zapytania jest następująca: zlicz te urządzenia, których *uptime* jest mniejszy od zadanego przez użytkownika progu (np. 900 sekund). Dzięki temu wygenerowana zostanie lista urządzeń, których nieprzerwany czas pracy jest krótszy od 15 minut. Należy jednak przy tym uwzględnić czas dodania (utworzenia) danego urządzenia do bazy – jeżeli urządzenie zostałoby dodane bardzo dawno temu (w stosunku do czasu dokonania ostatniego pomiaru), to wówczas wysoka częstość występowania w przedstawionym zapytaniu, nie świadczy o dużej jego awaryjności. Wynik zapytania 8.8 został zaprezentowany w postaci tabeli 8.9.

Z tabeli 8.9 wynika, że najbardziej awaryjnym (pod kątem krótkiego czasu nieustannej pracy) jest urządzenie *AP-Turnberry-1*, ponieważ aż 20 razy posiadało bardzo mały czas dzia-

Tabela 8.9: Wynik zapytania 8.8

czestosc	nazwa	utworzono
20	AP-Turnberry-1	2010-07-01 05:30:54
13	AP-BC-RedMtn-1	2010-07-28 22:24:05
11	AP-Luxor-1	2009-07-04 10:03:41
10	AP-LVNET-1	2010-07-13 21:07:40
8	AP-Carriage-House-1	2010-06-12 03:56:26
7	AP-Plaza-1	2010-01-12 20:00:26
6	AP-DonSmith-1	2009-11-09 19:02:05
4	AP-Hilton-1	2009-08-13 23:29:13
3	AP-Luxor-2	2009-11-09 22:26:34
2	AP-Plaza-2	2010-07-24 00:48:52
1	AP-Hilton-2	2010-07-13 19:30:09
1	AP-Primm-1	2010-03-19 19:39:27

łania, krótszy od 15 minut. Tezę tę potwierdza również kolumna *dodano*, symbolizująca datę i czas dodania urządzenia do bazy – występują bowiem punkty dostępowe, które zostały dodane znacząco wcześniej względem wspominanego *AP-Turnberry-1* jak *AP-Hilton-1* czy *AP-Luxor-2* (co oznacza, że pracują dłużej), a jednak ich czas nieprzerwanej pracy jest lepszy (ponieważ tylko trzy lub cztery razy wynosił on mniej niż 15 minut). Świadczy to o konieczności niezwłocznej wymiany tego urządzenia, co potencjalnie pozwoli na poprawę dostępności oferowanych usług (np. dostępu do internetu).

Eksperyment 7. Wyznaczenie optymalnych parametrów sterujących dla gęstościowych algorytmów grupowania, na podstawie zbioru ap_loss

Następny eksperyment dotyczy ustalenia optymalnych wartości parametrów dla algorytmu gęstościowego oraz tabel *device_aggreg* i *interface_aggreg*, wchodzących w skład zbioru *ap_loss*. Tabela *devices* została pominięta, ponieważ posiada prostą strukturę i relatywnie niewiele rekordów. Tabela *stations_aggreg* również nie jest uwzględniona w dalszych analizach, ponieważ jak przedstawiono podczas wstępnego opisu zawartych w niej danych za pomocą histogramów, trudno jest ostatecznie postawić rozsądny cel eksploracji biorąc pod uwagę przechowywane tam dane (oraz specyfikę atrybutów wchodzących w skład opisów obiektów).

Do wyznaczenia optymalnych parametrów sterujących dla zaimplementowanych algorytmów gęstościowych, wykorzystano heurystykę przedstawioną podczas omawiania eksperymentu numer trzy. W przypadku tabeli *device_aggreg*, do grupowania uwzględniono 4 atrybuty: identyfikator urządzenia (*device_id*), wykorzystanie procesora (*cpu_usages*), temperaturę (*temps*), zajętość pamięci (*memory_usages*). Odrzucono zatem atrybuty stanowiące klucz główny tabeli oraz atrybut (*uptime*) symbolizujący czas nieprzerwanej pracy urządzenia, ponieważ w procesie eksploracji skupiono się na relacji między temperaturą a wykorzystaniem zasobów sprzętowych (zgodnie z sugestią eksperta dziedzinowego). Natomiast podczas grupowania tabeli *interface_aggreg*, odrzucono zarówno atrybuty wchodzące w skład klucza głównego, jak również liczbę pakietów wysłanych (*tx_packets_rate*) i odebranych (*tx_packets_rate*).

Zdecydowano się nie uwzględniać liczby pakietów, ponieważ w strukturze tabeli znajdują się również pola *tx_bytes_rate* oraz *rx_bytes_rate*, symbolizujące kolejno liczbę wysłanych i odebranych bajtów. Naturalnie im więcej transmitowanych jest pakietów podczas transmisji, tym bardziej wzrasta wskaźnik *tx_bytes_rate*. Ponieważ jednak każdy pakiet może przenosić różną liczbę informacji (bajtów), sama ich liczba niewiele mówi o transmisji i bardziej odpowiednią wielkością do uwzględnienia jest właśnie transmitowana liczba bajtów. Niemniej liczba pakietów została za radą eksperta dziedzinowego pozostawiona w strukturze tabeli, ponieważ być może pozwoli na wyjaśnienie przyczyn zależności, odkrytych podczas analizy skupień. Niestety w procesie wyznaczania optymalnych parametrów dla algorytmów analizy skupień, problematyczny okazał się rozmiar wspomnianych zbiorów. Tabela *device_aggreg* agreguje bowiem 204258 obiektów, natomiast tabela *interface_aggreg* przechowuje ich 406799. Tak duża liczba obiektów znacząco wydłuża czas generowania pojedynczego grupowania (bez względu na zastosowany algorytm gęstościowy) – eksperymenty przerwano po 48 godzinach pracy metod gęstościowych. Długi czas oczekiwania na rezultaty (rzędu kilku dni czy nawet tygodni) bardzo często jest niedopuszczalny w zastosowaniach telekomunikacyjnych, gdzie niemalże codziennie realizowane są naprawy/wymiany urządzeń sieciowych czy też ogólnie modernizacja struktury. Jest to szczególnie istotne, jeżeli omawiane w ramach rozprawy techniki odkrywania wiedzy, miałyby zostać zastosowane w rzeczywistych systemach (np. monitorujących na bieżąco pracę sieci). Dlatego też autor rozprawy przyjął granicę 24 godzin, jako maksymalny czas wykonywania obliczeń dla zbiorów analizowanych w rozprawie. Jednakże by ten cel osiągnąć, należało wprowadzić mechanizmy próbkowania danych, pozwalające na ograniczenie wielkości zbioru źródłowego (dla zbiorów liczących powyżej 150 tysięcy obiektów). W pracy [48] autora rozprawy zostały przedstawione spotykane w literaturze przedmiotu podejścia stosowane do grupowania dużych wolumenów danych złożonych, wraz z przykładami algorytmów je wykorzystujących oraz zestawieniem wad i zalet konkretnej techniki. W przypadku pracy z rzeczywistymi zbiorami danych złożonych, metoda próbkowania powinna uwzględniać przede wszystkim ich dziedzinowy charakter i umożliwić reprezentowanie kluczowych aspektów danego zbioru. W związku z czym, niemożliwy do zastosowania jest losowy wybór obiektów, aż do osiągnięcia zadanej liczby, a także wybór wyłącznie pewnego procentu obiektów, znajdujących się na początkowych miejscach tabeli. Dlatego też autor rozprawy (biorąc pod uwagę specyfikę zbioru *ap_loss*) proponuje następującą metodę wyboru obiektów do analizy: należy wydobyć określony przez użytkownika procent rekordów tabeli (aby możliwe było sterowanie wielkością próbki) jednocześnie gwarantując, że dotyczą one wszystkich urządzeń sieciowych pamiętanych w danym zbiorze. Zostało to zrealizowane za pomocą zapytania SQL zaprezentowanego na listingu 8.9.

Listing 8.9: Proponowana metoda próbkowania obiektów do grupowania.

```
USE ap_loss

DECLARE @procent tinyint
DECLARE @device_id int
DECLARE @tableName NVARCHAR(200)
DECLARE @tempTableName NVARCHAR(200)
```

```
DECLARE @SQLString NVARCHAR(500)

SET @procent = 10
SET @tableName = 'device_aggreg'

SET @tempTableName = 'dbo.' + @tableName + '_temp'

IF OBJECT_ID(@tempTableName, 'U') IS NOT NULL BEGIN
   SET @SQLString = 'DROP TABLE ' + @tempTableName
   EXECUTE sp_executesql @SQLString
END

SET @SQLString = 'SELECT TOP 0 * INTO ' + @tempTableName + ' FROM '
 + @tableName
EXECUTE sp_executesql @SQLString

DECLARE k_id CURSOR READ_ONLY FOR
  SELECT id FROM devices

OPEN k_id
FETCH NEXT FROM k_id INTO @device_id
WHILE (@@fetch_status <> -1) BEGIN
  IF (@@fetch_status <> -2) BEGIN
    SET @SQLString = 'INSERT INTO ' + @tempTableName + ' SELECT TOP ('
    + CONVERT(NVARCHAR(20), @procent) + ') PERCENT * FROM '
    + @tableName + ' WHERE device_id = ' + CONVERT(NVARCHAR(20), @device_id)
    EXECUTE sp_executesql @SQLString
  END
  FETCH NEXT FROM k_id INTO @device_id
END
CLOSE k_id
DEALLOCATE k_id
```

Na początku zapytania 8.9 deklarowane są zmienne przechowujące procent obiektów, do jakiego należy zredukować wynikowy zbiór danych, nazwę tabeli z której dane będą pobierane (próbkowane), tymczasowy identyfikator analizowanego urządzenia, nazwę wynikowej tabeli oraz aktualną postać podzapytania. Rolą użytkownika jest ewentualna zmiana wartości zmiennej *procent* oraz *tableName*. Na potrzeby opisywanych w tej sekcji eksperymentów, wybranych zostało około 10% wszystkich rekordów z analizowanych tabel. Wartość ta została dobrana eksperymentalnie i gwarantuje, że eksperymenty związane z określeniem parametrów algorytmów analizy skupień, zakończą się do 24 godzin dla analizowanych zbiorów. Jednakże wartość ta może być dowolnie modyfikowana w zależności od ograniczeń czasowych i wielkości badanego zbioru. Następnie tworzona jest pusta tablica tymczasowa z przyrostkiem *_temp*, o strukturze zgodnej z tabelą źródłową. Dalsze postępowanie to stworzenie tzw. kursora, czyli mechanizmu, którego zadaniem jest szybkie pobranie wszystkich identyfikatorów, urządzeń przechowywanych w tabeli *devices*. Następnie rozpoczyna się pętla *while*, która w pierwszym kroku wybiera określony przez użytkownika (domyślnie 10) procent rekordów z tabeli źródłowej, dotyczących wyłącznie konkretnego urządzenia (o zadanym *device_id*). Jednakże po analizie wszystkich urządzeń, do tabeli tymczasowej wybieranych jest ostatecznie po 10% rekordów, opisujących

Tabela 8.10: Wyznaczanie optymalnych parametrów algorytmu gęstościowego na podstawie tabeli *device_aggreg_temp*

Eps	**MinPts**	**Liczba grup**	**Wartości izolowane**	**Pierwsza największa grupa**	**Druga największa grupa**	**Trzecia największa grupa**
0	1	20072	Brak	2	2	2
0	2	359	19713	2	2	2
0	3	0	20431	Brak	Brak	Brak
1	1	3701	Brak	1396	938	917
1	2	2136	1565	1396	938	917
1	3	1458	2921	1396	938	917
2	1	11	Brak	20421	1	1
2	2	1	10	20421	Brak	Brak
2	3	1	10	20421	Brak	Brak
3	1	1	Brak	20431	Brak	Brak
3	2	1	Brak	20431	Brak	Brak
3	3	1	Brak	20431	Brak	Brak
4	1	1	Brak	20431	Brak	Brak
4	2	1	Brak	20431	Brak	Brak
4	3	1	Brak	20431	Brak	Brak
5	1	1	Brak	20431	Brak	Brak
5	2	1	Brak	20431	Brak	Brak
5	3	1	Brak	20431	Brak	Brak

pracę każdego urządzenia występującego w tabeli źródłowej (*device_aggreg*).

Na podstawie zapytania 8.9 utworzono zatem dwie tabele *device_aggreg_temp* oraz *interface_aggreg_temp*, liczące odpowiednio 20431 oraz 40685 obiektów. Stanowią one podstawę wszystkich dalszych eksperymentów wykonywanych dla zbioru *ap_loss*. Wyniki procesu grupowania (z różnymi wartościami parametrów *Eps* i *MinPts* użytego algorytmu gęstościowego) dla *device_aggreg_temp* przedstawia tabela 8.10, natomiast dla zestawu danych *interface_aggreg_temp* wyniki przedstawia tabela 8.11.

Analizując wyniki zawarte w tabeli 8.10 można stwierdzić, że podobnie jak dla eksperymentu numer trzy, największy wpływ na strukturę grup ma zmiana promienia sąsiedztwa *Eps*. Jednakże w tym przypadku, parametr *MinPts* również ma spore oddziaływanie na liczbę utworzonych grup. Jest to szczególnie widoczne, gdy parametr *Eps* wynosi zero[16] lub jeden. Można zatem stwierdzić, że w rezultacie procesu analizy skupień powstaje dużo małolicznych grup. Identyczną interpretację można zastosować, badając zawartość tabeli 8.11. Zatem jako parametry optymalne dla algorytmu gęstościowego DBSCAN, zostały przyjęte w przypadku tabeli *device_aggreg_temp* wartości *Eps = 1*, *MinPts = 1*, natomiast dla tabeli *interface_aggreg_temp* *Eps = 2*, *MinPts = 1*.

[16]Taką sytuację należy interpretować następująco: obiekty muszą posiadać dokładnie te same wartości cech branych pod uwagę przy grupowaniu, by trafić do tego samego skupienia.

Tabela 8.11: Wyznaczanie optymalnych parametrów algorytmu gęstościowego, na podstawie tabeli *interface_aggreg_temp*

Eps	MinPts	Liczba grup	Wartości izolowane	Pierwsza największa grupa	Druga największa grupa	Trzecia największa grupa
0	1	31300	Brak	1524	1463	507
0	2	820	30480	1524	1463	507
0	3	326	31468	1524	1463	507
1	1	19815	Brak	2070	1524	1473
1	2	2744	17071	2070	1524	1473
1	3	854	20851	2070	1524	1473
2	1	15	Brak	33928	3637	3108
2	2	3	12	33928	3637	3108
2	3	3	12	33928	3637	3108
3	1	1	Brak	40685	Brak	Brak
3	2	1	Brak	40685	Brak	Brak
3	3	1	Brak	40685	Brak	Brak
4	1	1	Brak	40685	Brak	Brak
4	2	1	Brak	40685	Brak	Brak
4	3	1	Brak	40685	Brak	Brak
5	1	1	Brak	40685	Brak	Brak
5	2	1	Brak	40685	Brak	Brak
5	3	1	Brak	40685	Brak	Brak

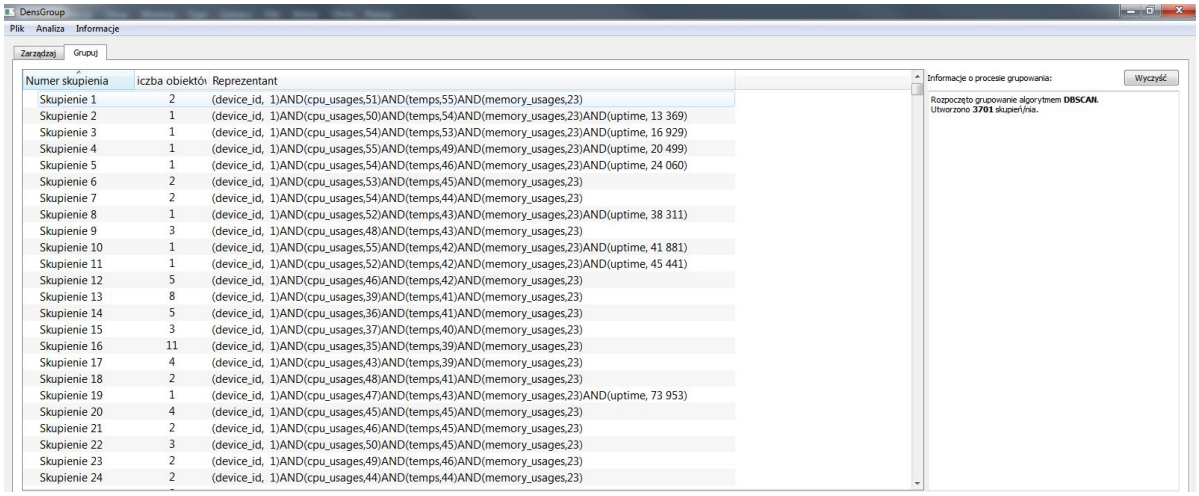
Eksperyment 8. Analiza przydatności zastosowania systemu *DensGroup* w procesie odkrywania wiedzy, na podstawie zbioru ap_loss

Celem ostatniego wykonanego eksperymentu, będzie potwierdzenie użyteczności narzędzia *DensGroup* do odkrywania korelacji między danymi w zbiorze *ap_loss* (ograniczonymi do 10% wszystkich instancji, zgodnie z przedstawioną wcześniej metodą próbkowania). Dokonano zatem dwóch grupowań (dotyczących tabel *device_aggreg* oraz *interface_aggreg*) algorytmem DBSCAN, dla ustalonych wcześniej wartości optymalnych parametrów promienia sąsiedztwa *Eps* i minimalnej liczby obiektów w grupie[17]. W zależności od liczności utworzonych skupień i stopnia skomplikowania ich reprezentantów, decydowano o uruchomieniu algorytmu aglomeracyjnego oraz wizualizacji za pomocą map prostokątów uzyskanej struktury. Wyniki grupowania dla tabeli *device_aggreg_temp* przedstawia rysunek 8.11.

Algorytm gęstościowy DBSCAN wygenerował tym razem niemalże 4 tysiące skupień. Jest to na tyle duża wartość, że utrudnia skuteczną analizę. Dlatego też zastosowano algorytm AHC, tym razem ustawiając liczbę grup na stałą wartość 80 skupień[18]. Technikę łączenia skupień ustawiono jako średnie wiązanie. Następnie utworzono wizualizację otrzymanej struktury

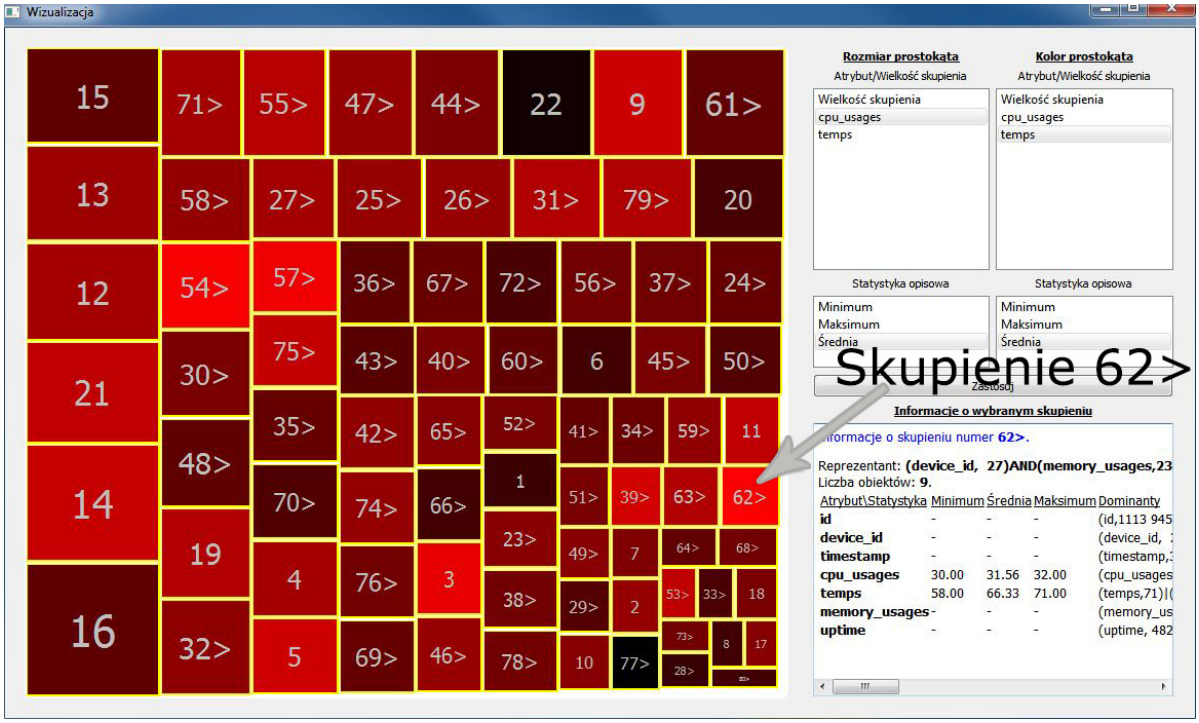
[17]Wybór atrybutów uwzględnianych podczas grupowania również nie uległ zmianie, względem informacji zamieszczonych w poprzedniej sekcji.

[18]Wartość została dobrana arbitralnie, aby znacząco zredukować liczbę grup i ułatwić późniejszą analizę, a także zaprezentować różne możliwości sterowania procesem grupowania hierarchicznego przez użytkownika.



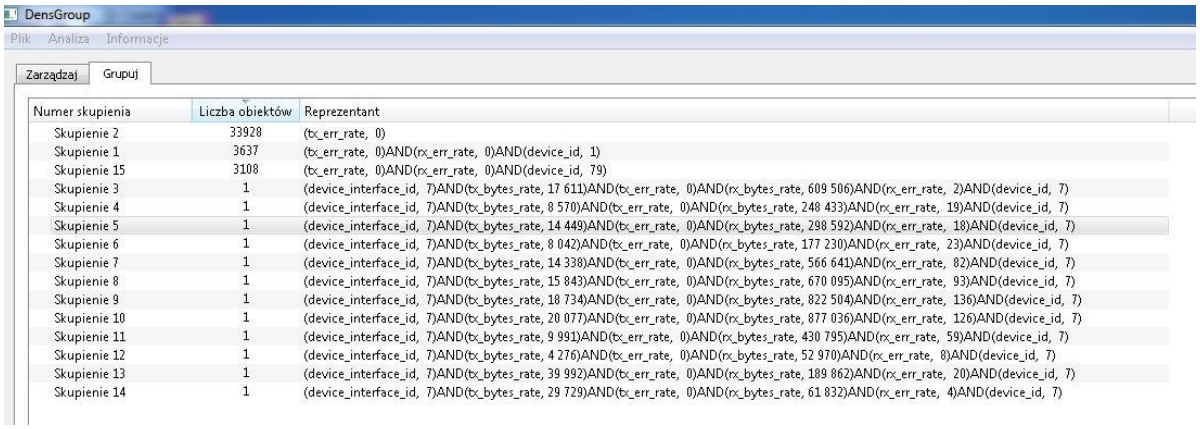
Rysunek 8.11: Wynik zastosowania algorytmu DBSCAN na tabeli *device_aggreg_temp*.
Źródło: Opracowanie własne

(wykorzystując technikę map prostokątów), co zostało zaprezentowane na rysunku 8.12. Na wizualizacji poszukiwano korelacji między niskim wykorzystaniem procesora a wysoką temperaturą. Zatem rozmiar prostokątów odpowiada obciążeniu procesora, natomiast ich kolor temperaturze.



Rysunek 8.12: Wizualizacja struktury hierarchicznej wygenerowanej na podstawie tabeli *device_aggreg_temp*.
Źródło: Opracowanie własne

Analizując wizualizację przedstawioną na rysunku 8.12, zwrócono uwagę na skupienie numer *62>*. Zawiera ono jedno urządzenie o identyfikatorze *27*, charakteryzujące się maksymalnym, zarejestrowanym obciążeniem procesora na poziomie zaledwie 32% i średnią temperaturą pracy powyżej 66 stopni. Identyfikator *27* odpowiada urządzeniu o nazwie *AP-Plaza-1*, które zostało również zwrócone jako wynik zapytania 8.5. Pozwala to zatem potwierdzić wcześniejsze spostrzeżenia i sugeruje, że wykryte urządzenie powinno być potencjalnie lepiej chłodzone.


DensGroup

Plik Analiza Informacje

Zarządzaj Grupuj

Numer skupienia	Liczba obiektów	Reprezentant
Skupienie 2	33928	(tx_err_rate, 0)
Skupienie 1	3637	(tx_err_rate, 0)AND(rx_err_rate, 0)AND(device_id, 1)
Skupienie 15	3108	(tx_err_rate, 0)AND(rx_err_rate, 0)AND(device_id, 79)
Skupienie 3	1	(device_interface_id, 7)AND(tx_bytes_rate, 17 611)AND(tx_err_rate, 0)AND(rx_bytes_rate, 609 506)AND(rx_err_rate, 2)AND(device_id, 7)
Skupienie 4	1	(device_interface_id, 7)AND(tx_bytes_rate, 8 570)AND(tx_err_rate, 0)AND(rx_bytes_rate, 248 433)AND(rx_err_rate, 19)AND(device_id, 7)
Skupienie 5	1	(device_interface_id, 7)AND(tx_bytes_rate, 14 449)AND(tx_err_rate, 0)AND(rx_bytes_rate, 298 592)AND(rx_err_rate, 18)AND(device_id, 7)
Skupienie 6	1	(device_interface_id, 7)AND(tx_bytes_rate, 8 042)AND(tx_err_rate, 0)AND(rx_bytes_rate, 177 230)AND(rx_err_rate, 23)AND(device_id, 7)
Skupienie 7	1	(device_interface_id, 7)AND(tx_bytes_rate, 14 338)AND(tx_err_rate, 0)AND(rx_bytes_rate, 566 641)AND(rx_err_rate, 82)AND(device_id, 7)
Skupienie 8	1	(device_interface_id, 7)AND(tx_bytes_rate, 15 843)AND(tx_err_rate, 0)AND(rx_bytes_rate, 670 095)AND(rx_err_rate, 93)AND(device_id, 7)
Skupienie 9	1	(device_interface_id, 7)AND(tx_bytes_rate, 18 734)AND(tx_err_rate, 0)AND(rx_bytes_rate, 822 504)AND(rx_err_rate, 136)AND(device_id, 7)
Skupienie 10	1	(device_interface_id, 7)AND(tx_bytes_rate, 20 077)AND(tx_err_rate, 0)AND(rx_bytes_rate, 877 036)AND(rx_err_rate, 126)AND(device_id, 7)
Skupienie 11	1	(device_interface_id, 7)AND(tx_bytes_rate, 9 991)AND(tx_err_rate, 0)AND(rx_bytes_rate, 430 795)AND(rx_err_rate, 59)AND(device_id, 7)
Skupienie 12	1	(device_interface_id, 7)AND(tx_bytes_rate, 4 276)AND(tx_err_rate, 0)AND(rx_bytes_rate, 52 970)AND(rx_err_rate, 8)AND(device_id, 7)
Skupienie 13	1	(device_interface_id, 7)AND(tx_bytes_rate, 39 992)AND(tx_err_rate, 0)AND(rx_bytes_rate, 189 862)AND(rx_err_rate, 20)AND(device_id, 7)
Skupienie 14	1	(device_interface_id, 7)AND(tx_bytes_rate, 29 729)AND(tx_err_rate, 0)AND(rx_bytes_rate, 61 832)AND(rx_err_rate, 4)AND(device_id, 7)

Rysunek 8.13: Wynik zastosowania algorytmu DBSCAN na tabeli *interface_aggreg_temp*.
Źródło: Opracowanie własne

Podobne analizy przeprowadzono dla tabeli *interface_aggreg_temp*. Wynik grupowania, dla ustalonych w poprzedniej sekcji parametrów *Eps = 2*, *MinPts = 1*, przedstawia rysunek 8.13. Tym razem nie było potrzeby korzystania z algorytmu aglomeracyjnego czy też nawet wizualizacji, ponieważ w wyniku grupowania uzyskano 15 skupień, co jest wielkością możliwą do szybkiej analizy. Już na podstawie samego opisu reprezentantów grup, można odkryć ciekawą zależność. Otóż skupienia jednoelementowe o numerach od 3 do 14 są na tyle niepodobne do pozostałych, że zostały wyróżnione przez algorytm gęstościowy. Analiza opisu reprezentantów wspomnianych grup wykazała, że dotyczą one urządzenia o identyfikatorze *7* (i takim samym identyfikatorze interfejsu). Urządzenie to nosi nazwę *AP-Hilton-1* i charakteryzuje się wysoką liczbą błędów odbioru. Zostało ono również zwrócone jako wynik zapytania 8.7, co stanowi dodatkowe potwierdzenie wadliwej pracy tego punktu dostępowego.

8.3 Podsumowanie uzyskanych wyników

W wyniku przeprowadzonych eksperymentów udało się zrealizować główny cel rozprawy, jakim jest efektywne wydobywanie wiedzy z danych złożonych. Wszystkie analizy dotyczyły dwóch rzeczywistych zbiorów dotyczących zagadnień telekomunikacyjnych i sieciowych, jednakże zaprezentowane metody wydobywania wiedzy mogą zostać zastosowane do zbiorów o dowolnej tematyce.

Eksperymenty 1 i 5 potwierdziły przydatność wykorzystania histogramów, jako metody opisu danych, w procesie odkrywania wiedzy. Na ich podstawie zapoznano się ze strukturą zbiorów

danych[19], jak również potwierdzono obecność pewnych korelacji np. między datą a liczbą zarejestrowanych zdarzeń (związanych z pracą urządzeń nadawczo-odbiorczych). Ponadto tego typu techniki, w uproszczony sposób, mogą zasugerować na jakich atrybutach skupić dalsze analizy. Należy jednak nadmienić, że wykresy częstości dobrze sprawdzają się przede wszystkim dla cech jakościowych (nominalnych). Jest to ich istotne ograniczenie.

Eksperymenty numer 2 i 6 zweryfikowały możliwość zastosowania klauzul grupujących i funkcji agregujących języka SQL, celem znalezienia korelacji między obiektami w analizowanych zbiorach danych złożonych. Ponadto, wszystkie przedstawione zapytania mogą zostać bezpośrednio wdrożone w istniejące, automatyczne systemy monitorujące prace urządzeń sieciowych, ponieważ charakteryzuje je krótki czas wykonania (rzędu kilkudziesięciu sekund). Nie bez znaczenia jest również fakt, że do ich realizacji nie potrzeba żadnego dodatkowego oprogramowania – dane źródłowe i tak przechowywane są na serwerze w relacyjnej bazie danych.

Eksperymenty 3 i 7 polegały na wyborze optymalnych wartości parametrów startowych dla algorytmów gęstościowych. Zaproponowano prostą heurystykę doboru tych wielkości, jak również efektywną obliczeniowo technikę próbkowania danych, uwzględniającą ich dziedzinowy charakter.

Analiza wyników dostarczonych przez eksperymenty cztery i osiem pozwoliła potwierdzić przydatność autorskiego systemu *DensGroup* oraz zastosowanych algorytmów analizy skupień w procesie wydobywania wiedzy z rzeczywistych zbiorów danych złożonych. System *DensGroup* pozwolił bowiem na wygenerowanie nowej wiedzy (w postaci zidentyfikowania urządzenia nadawczo-odbiorczego wymagającego pilnie uwagi, nie wykrytego innymi metodami) jak również na potwierdzenie części wniosków z poprzednich eksperymentów. Eksperymentalnie pokazano przydatność zaimplementowanej techniki wizualizacji w połączeniu ze statystykami opisu danych.

Autor w ramach rozprawy przeprowadził szereg innych eksperymentów, opublikowanych następnie w recenzowanych publikacjach o zasięgu krajowym jak i międzynarodowym, których wykaz uwzględniono w dodatku B niniejszej pracy. Pozwoliły one na doprecyzowanie celów rozprawy oraz zapoznanie się z aktualnym stanem wiedzy na temat grupowania i wizualizacji danych złożonych.

[19] W niniejszej rozprawie zaprezentowano jedynie wykresy najciekawsze z punktu widzenia wydobywania wiedzy, jednakże jak zaznaczono wcześniej, histogramy wygenerowano dla każdego atrybutu.

Rozdział 9

Podsumowanie

W obecnej chwili notuje się wzrost zapotrzebowania na urządzenia mobilne, a w szczególności telefony komórkowe i tablety, mające zapewnić dostęp do kultury i rozrywki, ale również oferujące możliwość edukacji czy pracy zdalnej. Malejące ceny tego typu sprzętu elektronicznego przyczyniły się dodatkowo do jego rozpowszechnienia wśród użytkowników prywatnych, ale przede wszystkim wśród wielu przedsiębiorstw. W celu zaspokojenia rosnących wymagań wszystkich użytkowników, wprowadzane są coraz to nowsze usługi i pakiety transmisji danych, w tym dostęp do cyfrowej telewizji i wieloosobowych wideokonferencji. Usługi informacyjne automatycznie dostosowują się do zmiennych potrzeb biznesu, a to wymaga bardziej efektywnego zarządzania siecią, zwłaszcza w dużych przedsiębiorstwach i korporacjach. Prowadzi to do agregacji wielu różnych narzędzi, w celu zapewnienia ciągłości dostępu do oferowanych usług i ich wysokiej jakości.

Niestety monitorowanie i utrzymanie skomplikowanych sieci telekomunikacyjnych jest zadaniem trudnym, przez co również często kosztownym. Problem stanowią nie tylko zmieniające się wymogi użytkowników, czy nierównomierny rozkład obciążenia sieci, ale również niekompatybilność urządzeń sieciowych względem siebie. Dlatego też obserwuje się podwyższony popyt na oprogramowanie, które potrafi automatycznie zbierać i przetwarzać dane pochodzące z różnych źródeł i urządzeń sieciowych oraz na tej podstawie odpowiednio reagować np. przez wysłanie monitu do administratora. Oprócz informacji o awarii, istotniejszym czynnikiem z punktu widzenia usługodawcy, jest przyczyna powstałego problemu. Niestety w obliczu natłoku zgromadzonych danych, przechowywanych często w różnych źródłach i formatach, nie jest możliwa ich dogłębna analiza (celem wykrycia przyczyn problemów) przy wykorzystaniu tradycyjnych technik statystycznych. Dlatego też poszukuje się metod odkrywających zależności, trendy czy relacje, które mogą zostać zaaplikowane do dużych zbiorów danych, w celu wygenerowania nowej i poprawnej wiedzy (np. na temat pracy urządzeń czy awarii sieci). Analiza takich metod na danych rzeczywistych jest przedmiotem tematyki niniejszej rozprawy.

By analiza danych była efektywna, należy zadbać wcześniej o odpowiednie przygotowanie danych. Często bowiem zastosowanie określonej metody eksploracji danych, uzależnione jest od konkretnej postaci zbioru danych. W tym celu dane trzeba np. poddać transformacji do

określonego formatu, dyskretyzacji czy wręcz operacji czyszczenia z obserwacji błędnych, bądź z niekompletnym opisem, które powodują, że pewna metoda analizy w ogóle nie może być zastosowana, póki takie braki w danych mają miejsce. Ze względu na fakt, że bardzo często bazy danych i narzędzia przez nie udostępniane nie są wystarczającymi, gdy celem analizy jest odkrywanie nowej wiedzy o badanej dziedzinie, niniejsza praca poświęcona jest zagadnieniom analizy danych opartej na statystyce oraz metodom eksploracji danych.

Spośród wielu technik eksploracji danych zdecydowano się wybrać analizę skupień [7] i to właśnie wszystkie aspekty realizacji tej techniki, w odniesieniu do danych złożonych, są podstawą niniejszej rozprawy. Szczególna uwaga została poświęcona algorytmom gęstościowym, które to charakteryzują się m.in. możliwością odkrywania skupień o różnej strukturze czy odpornością na występowanie obiektów izolowanych, co jest bardzo istotne podczas przetwarzania danych rzeczywistych. W pracy omówiono także metodę optymalnego doboru parametrów dla procesu grupowania, próbkowania danych oraz opisu i wizualizacji utworzonej struktury złożonych grup. Ze względu na fakt, że przedstawione w rozprawie eksperymenty dotyczyły przetwarzania danych rzeczywistych, wszystkie wnioski zostały skonsultowane i potwierdzone przez ekspertów dziedzinowych, odpowiedzialnych za administrację tymi zbiorami.

9.1 Szczegółowe wyniki rozprawy

W rozprawie przeanalizowano oraz wykorzystano wybrane algorytmy analizy skupień, jako metodę wydobywania wiedzy z dużych wolumenów danych złożonych. Przeprowadzono analizę problemu oraz zaproponowano metodę doboru optymalnych parametrów startowych dla algorytmów gęstościowych. Stworzono technikę próbkowania danych, uwzględniającą ich dziedzinowy charakter. Zaakcentowano oraz dowiedziono eksperymentalnie przydatność technik przygotowania danych, jako narzędzia ekstrakcji użytecznej wiedzy. Dokonano przeglądu porównawczego istniejącego oprogramowania do analizy danych, szczególną uwagę poświęcając możliwościom grupowania i wizualizacji struktury grup, uzasadniając tym samym konieczność stworzenia autorskiego systemu wydobywania wiedzy *DensGroup*. Skuteczność analizy skupień w dużej mierze zależy od jakości opisów reprezentantów skupień. Reprezentant grupy, nieadekwatny do jej zawartości, uniemożliwi efektywne przeszukiwanie struktury skupień, a przez to wpłynie na błędy w wyszukiwaniu informacji w danych. Z tego względu, metodom reprezentacji skupień poświęcono w ramach rozprawy także sporo uwagi. Zdefiniowano cztery metody opisu wygenerowanych skupień oraz podano przyczyny wyboru techniki opartej o koniunkcję deskryptorów. Przeprowadzono przegląd literaturowy graficznych metod reprezentacji skupień, z wykazaniem zalet techniki map prostokątów, wykorzystywanej w systemie *DensGroup*. Dowiedziono eksperymentalnie przydatność narzędzia *DensGroup* oraz innych opisanych technik wydobywania wiedzy.

We wszystkich przeprowadzonych rozważaniach i eksperymentach wykorzystano dwa zestawy rzeczywistych danych, związanych z funkcjonowaniem sieci telekomunikacyjnych. Należy również zaznaczyć, że zdecydowaną większość procedur, opisanych podczas wykonywania eksperymentów, można bezpośrednio zastosować w przemyśle (np. w systemach monitorujących pracę urządzeń sieciowych).

Wyniki przeprowadzonych badań sugerują, że wydobywanie wiedzy z danych złożonych jest procesem wieloetapowym, który dla rzeczywistych zbiorów powinien być przeprowadzany przy ścisłej współpracy z ekspertem dziedzinowym. Wykorzystanie możliwości wizualizacji struktury grup oraz analiza powstałych (w wyniku zastosowania opisanych algorytmów) skupień, pozwoli na ekstrakcję nowej, użytecznej wiedzy, która w przedstawionym przypadku może przyczynić się do lepszego wykorzystania i monitorowania dostępnych zasobów sieci telekomunikacyjnej.

Bibliografia

[1] J. Abonyi, B. Feil. *Cluster Analysis for Data Mining and System Identification.* Birkhäuser Basel, Niemcy, 2007. [cytowanie na str. 17]

[2] E. Achtert, C. Böhm, H.-P. Kriegel, P. Kröger. *Online Hierarchical Clustering in a Data Warehouse Environment.* PROC. 5TH IEEE INT. CONF. ON DATA MINING (ICDM'05): 10–17, Washington, DC, USA, 2005. [cytowanie na str. 77]

[3] M. Ankerst. *Visual Data Mining.* Praca doktorska, Faculty of Mathematics and Computer Science, University of Munich, 2001. [cytowanie na str. 2, 79, 80, 81, 82]

[4] M. Ankerst, M. M. Breunig, H.-P. Kriegel, J. Sander. *OPTICS: Ordering Points To Identify the Clustering Structure.* SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, Philadelphia, USA, 1999. [cytowanie na str. 69, 70, 71, 74, 75, 77, 86, 108, 128]

[5] M. Balzer, O. Deussen. *Voronoi Treemaps.* Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization, INFOVIS '05, Washington, DC, USA, 2005. [cytowanie na str. 96]

[6] P. Berka, J. Rauch, D. A. Zighed. *Data Mining and Medical Knowledge Management: Cases and Applications.* Information Science Reference - Imprint of: IGI Publishing, Hershey, USA, 2009. [cytowanie na str. 1]

[7] M. W. Berry, M. Browne, redaktorzy. *Lecture Notes in Data Mining.* World Scientific Publishing Co. Pte. Ltd, Singapur, 2006. [cytowanie na str. 2, 16, 58, 150]

[8] M. R. Berthold, C. Borgelt, F. Höppner, F. Klawonn. *Guide to Intelligent Data Analysis: How to Intelligently Make Sense of Real Data.* Springer-Verlag, Londyn, Anglia, 2010. [cytowanie na str. 35, 49, 50, 52]

[9] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, K. Thiel, B. Wiswedel. *KNIME - the Konstanz information miner: version 2.0 and beyond.* SIGKDD Explor. Newsl., 11(1):26–31, 2009. [cytowanie na str. 33]

[10] T. Bladh, D. A. Carr, J. Scholl. *Extending Tree-Maps to Three Dimensions: A Comparative Study.* Proceedings of the 6th Asia-Pacific Conference on Computer-Human Interaction (APCHI 2004):50–59. Springer Verlag, 2004. [cytowanie na str. 97]

[11] M. Bruls, K. Huizing, J. van Wijk. *Squarified Treemaps.* Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization:3–42. Press, 1999. [cytowanie na str. 2, 93, 94, 98, 100]

[12] S. Chakrabarti, E. Cox, E. Frank, R. H. Güting, J. Han, X. Jiang, M. Kamber, S. S. Lightstone, T. P. Nadeau, R. E. Neapolitan, D. Pyle, M. Refaat, M. Schneider, T. J. Teorey, I. H. Witten. *Data Mining: Know It All.* Morgan Kaufmann Publishers, USA, 2008. [cytowanie na str. 43, 44, 45, 49, 53]

[13] J. Chustecki. *Vademecum teleinformatyka – tom 1.* Wydawnictwo IDG, 1999. [cytowanie na str. 6, 7, 8, 137, 159]

[14] T. Dasu, T. Johnson. *Exploratory Data Mining and Data Cleaning.* John Wiley & Sons, Inc., USA, 2003. [cytowanie na str. 41, 42, 43, 49]

[15] B. Engdahl. *Ordered and Unordered Treemap Algorithms and Their Applications on Handheld Devices.* Praca magisterska, Royal Institute of Technology, Sztokholm, Szwecja, 2005. dostępny w internecie: http://www.nada.kth.se/utbildning/grukth/exjobb/rapportlistor/2005/rapporter05/engdahl_bjorn_05033.pdf. [cytowanie na str. 93, 94, 96, 98]

[16] M. Ester, K. Ester, H.-P. Sander, Jorg, Sander, X. Xiaowei. *A density-based algorithm for discovering clusters in large spatial databases with noise.* Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, USA, 1996. [cytowanie na str. 66, 67, 130]

[17] U. Fayyad, G. G. Grinstein, A. Wierse, redaktorzy. *Information visualization in data mining and knowledge discovery.* Morgan Kaufmann Publishers Inc., 2002. [cytowanie na str. 84]

[18] D. Felcenloben. *Geoinformacja - wprowadzenie do systemów organizacji danych i wiedzy.* Wydawnictwo Gall, 2011. [cytowanie na str. 81]

[19] G. Gan, C. Ma, J. Wu. *Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Probability).* SIAM, Society for Industrial and Applied Mathematics, 2007. [cytowanie na str. 28, 32, 42, 109]

[20] F. Gorunescu. *Data Mining: Concepts, Models and Techniques.* Springer-Verlag, Berlin, Niemcy, 2011. [cytowanie na str. 41, 42, 43, 44, 46]

[21] M. Halkidi, Y. Batistakis, M. Vazirgiannis. *On Clustering Validation Techniques.* Kluwer Academic Publishers, 17(2-3):107–145, USA, 2001 [cytowanie na str. 28]

[22] J. Han, M. Kamber, J. Pei. *Data Mining: Concepts and Techniques.* Morgan Kaufmann Publishers, USA, 2012. [cytowanie na str. 16, 45, 46, 48, 53, 108, 114]

[23] D. Hand, H. Mannila, P. Smyth. *Eksploracja danych.* Wydawnictwo Naukowo-Techniczne, Warszawa, 2005. [cytowanie na str. 29, 30, 49, 109, 159]

[24] J. Herrero, A. Valencia, J. Dopazo. *A hierarchical unsupervised growing neural network for clustering gene expression patterns.* Bioinformatics, 17(1):126–136, 2001. [cytowanie na str. 34]

[25] T. Hill, P. Lewicki. *Statistics: Methods and Applications.* StatSoft, Inc., USA, 2006. [cytowanie na str. 51]

[26] T. Honkela, S. Kaski, K. Lagus, T. Kohonen. *WEBSOM - Self-Organizing Maps of Document Collections.* Neurocomputing 21:101–117, 1997. Dostępny w Internecie: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.98.3295&rep=rep1&type=pdf. [cytowanie na str. 59]

[27] ICONICS. *PLANTCockpit Technical Report – State of play and state of the art of flexible visualization frameworks.* Raport instytutowy, ICONICS, 2012. dostępny w internecie: http://www.plantcockpit.eu/fileadmin/PLANTCOCKPIT/user_upload/T6.1_PLANTCockpit_TechnicalReport_Visualization_Framework.pdf. [cytowanie na str. 89]

[28] A. K. Jain, N. M. Murty, P. J. Flynn. *Data clustering: A review.* ACM Comput. Surv., 31(3):264–323, 1999. [cytowanie na str. 16]

[29] T. Jain, S. Aggarwal, M. Trehan, V. Ohri. *Business Statistics.* Rahul Jain V.K. India Enterprises, 2011. [cytowanie na str. 49]

[30] B. Johnson, B. Shneiderman. Tree-Maps: a space-filling approach to the visualization of hierarchical information structures. *Proceedings of the 2nd conference on Visualization '91*, VIS '91:284–291, IEEE Computer Society Press, Los Alamitos, USA, 1991. [cytowanie na str. 94]

[31] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, G. Melancon. *Visual analytics: Definition, process, and challenges.* Information Visualization: Human-Centered Issues and Perspectives:154–175, 2008. [cytowanie na str. 79, 80]

[32] D. A. Keim. *Information Visualization and Visual Data Mining.* IEEE Transactions on Visualization and Computer Graphics 8:1–8, 2003. [cytowanie na str. 80, 81, 82]

[33] J. Korhonen. *Introduction to 3G Mobile Communications.* Artech House, Inc., Norwood, USA, 2001. [cytowanie na str. 7]

[34] J. Koronacki, J. Ćwik. *Statystyczne systemy uczące się.* Exit, Warszawa, 2008. [cytowanie na str. 57, 58, 59]

[35] J. Koronacki, J. Mielniczuk. *Statystyka dla studentów kierunków technicznych i przyrodniczych.* Wydawnictwa Naukowo-Techniczne, Warszawa, 2009. [cytowanie na str. 43, 47]

[36] A. Krause, M. O'Connell, redaktorzy. *A Picture is Worth a Thousand Tables: Graphics in Life Sciences.* Springer, New York, 2012. [cytowanie na str. 85]

[37] H.-P. Kriegel, P. Kröger, I. Gotlibovich. *Incremental OPTICS: Efficient Computation of Updates in a Hierarchical Cluster Ordering.* In 5th Int. Conf. on Data Warehousing and Knowledge Discovery:224–233, Springer, 2003. [cytowanie na str. 77]

[38] J. Lamping, R. Rao, P. Pirolli. *A focus+context technique based on hyperbolic geometry for visualizing large hierarchies.* Proceedings of the SIGCHI Conference on Human Factors in Computing Systems CHI '95, ACM Press/Addison-Wesley Publishing Co, USA, 1995. [cytowanie na str. 89]

[39] R. Mazza. *Introduction to Information Visualization.* Springer Publishing Company, Incorporated, 2009. [cytowanie na str. 89]

[40] A. Mucherino, P. J. Papajorgji, P. M. Pardalos. *Data Mining in Agriculture.* Springer, New York, USA, 2009. [cytowanie na str. 84]

[41] N. E. Nadar. *Statistics.* PHI Learning Private Limited, 2011. [cytowanie na str. 43, 48]

[42] A. Nanopoulos, A. Papadopoulos, Y. Theodoridis. *R-Trees: Theory and Applications.* Springer, New York, USA, 2006. [cytowanie na str. 66]

[43] A. Nowak-Brzezińska, T. Jach. *Wybrane aspekty wnioskowania w systemach z wiedzą niepełną. Studia Informatica*, 33 (2A):465–477, Wydawnictwo Politechniki Śląskiej, 2012. [cytowanie na str. 58]

[44] A. Nowak-Brzezińska, T. Jach, T. Xięski. *Analiza hierarchicznych i niehierarchicznych algorytmów grupowania dla dokumentów tekstowych. Studia Informatica*, Zeszyty Naukowe Politechniki Śląskiej, 30 (2A):245–258, Wydawnictwo Politechniki Śląskiej, 2009. [cytowanie na str. 58]

[45] A. Nowak-Brzezińska, T. Jach, T. Xięski. *Finding a relevant document in the clusters of documents' characteristics.* Intelligent Information Systems 2010:273–283, 2010. [cytowanie na str. 163]

[46] A. Nowak-Brzezińska, T. Xięski. Wybór algorytmu grupowania a efektywność wyszukiwania dokumentów. *Studia Informatica*, 31(2A):147–162, Wydawnictwo Politechniki Śląskiej, 2010. [cytowanie na str. 163]

[47] A. Nowak-Brzezińska, T. Xięski. Grupowanie danych złożonych. *Studia Informatica*, 32(2A):391–402, Wydawnictwo Politechniki Śląskiej, 2011. [cytowanie na str. 54]

[48] A. Nowak-Brzezińska, T. Xięski. Gęstościowa metoda grupowania i wizualizacji danych złożonych. *Studia Informatica*, 33(2A):453–464, Wydawnictwo Politechniki Śląskiej, 2012. [cytowanie na str. 58, 142, 163]

[49] A. Nowak-Brzezińska, T. Xięski. Metody reprezentacji danych złożonych. *Studia Informatica*, 34(2A):215–226, Wydawnictwo Politechniki Śląskiej, 2013. [cytowanie na str. 53, 83, 163]

[50] K. Onak, A. Sidiropoulos. *Circular partitions with applications to visualization and embeddings.* Proceedings of the twenty-fourth annual symposium on Computational geometry:28–37, New York, USA, 2008. ACM. [cytowanie na str. 96]

[51] C. Reimann, P. Filzmoser, R. Garrett, R. Dutter. *Statistical Data Analysis Explained: Applied Environmental Statistics with R.* John Wiley & Sons, Anglia, 2008. [cytowanie na str. 51, 53]

[52] J. Rekimoto, M. Green. *The Information Cube: Using Transparency in 3D Information Visualization.* Proceedings of the Third Annual Workshop on Information Technologies & Systems (WITS'93):125–132, 1993. [cytowanie na str. 97]

[53] G. Robertson, J. Mackinlay, S. Card. *Cone Trees: animated 3D visualizations of hierarchical information.* Proceedings of the SIGCHI Conference on Human Factors in Computing Systems:189–194, ACM, New York, USA, 1991. [cytowanie na str. 88, 89]

[54] F. Rossi. *Visual Data Mining and Machine Learning.* ESANN'2006 proceedings - European Symposium on Artificial Neural Networks:251–264, 2006. [cytowanie na str. 80]

[55] G. Salton. *Automatic Information Organization and Retreival.* McGraw-Hill, New York, USA, 1975. [cytowanie na str. 59]

[56] C. Sammut, G. I. Webb, redaktorzy. *Encyclopedia of Machine Learning.* Springer Publishing Company, Incorporated, USA, 2011. [cytowanie na str. 159]

[57] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization.* John Wiley & Sons, Inc., USA, 1992. [cytowanie na str. 50]

[58] B. Shneiderman. *Tree visualization with tree-maps: 2-d space-filling approach.* ACM Transactions on Graphics, 11(1):92–99, 1992. [cytowanie na str. 93]

[59] B. Shneiderman, M. Wattenberg. *Ordered Treemap Layouts.* Proceedings of the IEEE Symposium on Information Visualization 2001, INFOVIS '01, IEEE Computer Society, Washington, USA, 2001. [cytowanie na str. 94, 95]

[60] A. Siegel. *Practical Business Statistics.* Academic Press, USA, 2012. [cytowanie na str. 51]

[61] S. J. Simoff, M. H. Böhlen, A. Mazeika, redaktorzy. *Visual Data Mining. Theory, Techniques and Tools for Visual Analytics.* Springer Berlin Heidelberg, 2008. [cytowanie na str. 80]

[62] J. Stasko. *An evaluation of space-filling information visualizations for depicting hierarchical structures*. Int. J. Hum.-Comput. Stud., 53(5):663–694, 2000. [cytowanie na str. 90]

[63] J. B. Strother, J. M. Ulijn, Z. Fazal. *Information Overload: An International Challenge for Professional Engineers and Technical Communicators*. Wiley-IEEE Press, USA, 2012. [cytowanie na str. 79]

[64] P.-N. Tan, M. Steinbach, V. Kumar. *Introduction to Data Mining*. Addison-Wesley, USA, 2006. [cytowanie na str. 57, 66, 83]

[65] J. J. Thomas, K. A. Cook, redaktorzy. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society Press, 2005. [cytowanie na str. 80]

[66] S. Tufféry. *Data Mining and Statistics for Decision Making*. John Wiley & Sons, Inc., Anglia, 2011. [cytowanie na str. 16]

[67] R. Vliegen, J. J. van Wijk, E.-J. van der Linden. *Visualizing Business Data with Generalized Treemaps*. IEEE Transactions on Visualization and Computer Graphics, 12(5):789–796, 2006. [cytowanie na str. 94]

[68] A. Wakulicz-Deja, A. Nowak-Brzezińska, T. Xięski. *Efficiency of complex data clustering*. Lecture Notes in Computer Science, 6954:636–641, Springer Berlin Heidelberg, 2011. [cytowanie na str. 53, 54, 163]

[69] A. Wakulicz-Deja, A. Nowak-Brzezińska, T. Xięski. *Density-based method for clustering and visualization of complex data*. Lecture Notes in Computer Science:142–149, Springer Berlin Heidelberg, 2012. [cytowanie na str. 163]

[70] G. Williams. *Data Mining with Rattle and R The Art of Excavating Data for Knowledge Discovery*. Springer Science+Business Media, LLC, 2011. [cytowanie na str. 37]

[71] T. Xięski. *Analiza gęstościowych algorytmów grupowania*. Praca magisterska, Uniwersytet Śląski, Sosnowiec, Polska, 2010. [cytowanie na str. 28, 86, 130]

[72] T. Xięski. *Grupowanie danych złożonych*. A. Wakulicz-Deja, redaktor, Systemy wspomagania decyzji:147–155, Wydawnictwo Uniwersytetu Śląskiego, Sosnowiec, 2011. [cytowanie na str. 6, 9, 163]

[73] T. Xięski. *Wybrane aspekty grupowania danych złożonych*. A. Wakulicz-Deja, redaktor, Systemy wspomagania decyzji:205–213, Wydawnictwo Uniwersytetu Śląskiego, Sosnowiec, 2011. [cytowanie na str. 163]

[74] T. Xięski. *Metody reprezentacji danych złożonych*. A. Wakulicz-Deja, redaktor, Systemy wspomagania decyzji, Wydawnictwo Uniwersytetu Śląskiego, Sosnowiec, 2013 [W druku]. [cytowanie na str. 109, 163]

[75] F. W. Young, P. M. Valero-Mora, M. Friendly. *Visual Statistics: Seeing Data with Dynamic Interactive Graphics*. John Wiley & Sons, Inc., USA, 2006. [cytowanie na str. 50, 53]

[76] Ankieta internetowa serwisu KDnuggets odnośnie wykorzystywanych narzędzi analizy danych. Dostępny w Internecie: http://www.kdnuggets.com/polls/2012/analytics-data-mining-big-data-software.html. [cytowanie na str. 22, 23]

[77] Broszura informacyjna opisująca możliwości oprogramowania SAS Enteprise Miner. Dostępny w Internecie: http://www.sas.com/offices/europe/poland/software/SAS_Enterprise_Miner.pdf. [cytowanie na str. 24, 25]

[78] Broszura informacyjna opisująca możliwości oprogramowania STATISTICA Data Miner. Dostępny w Internecie: http://www.statsoft.pl/dataminer.html. [cytowanie na str. 32]

[79] Cisco. The Zettabyte Era—Trends and Analysis. Dostępny w Internecie: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/VNI_Hyperconnectivity_WP.pdf. [cytowanie na str. 5]

[80] Dokumentacja biblioteki JFreeChart. Dostępny w Internecie: http://www.jfree.org/jfreechart/api/javadoc/index.html. [cytowanie na str. 35]

[81] Dokumentacja programu IBM SPSS Modeler 15. Dostępny w Internecie: http://www-01.ibm.com/support/docview.wss?uid=swg27023172. [cytowanie na str. 27]

[82] Dokumentacja programu MS SQL Server 7.0 OLAP Services. Dostępny w Internecie: http://technet.microsoft.com/en-us/library/cc966398.aspx. [cytowanie na str. 29]

[83] Dokumentacja techniczna algorytmów grupowania dostępnych w Microsoft Analysis Services. Dostępny w Internecie: http://technet.microsoft.com/pl-pl/library/cc280445.aspx. [cytowanie na str. 30]

[84] Dokumentacja techniczna algorytmów zastosowanych w programie RapidMiner. Dostępny w Internecie: http://docs.rapid-i.com/files/rapidminer/RapidMiner_OperatorReference_en.pdf. [cytowanie na str. 38]

[85] Fragment dokumentacji do systemu SAS Enteprise Miner odnośnie technik grupowania danych. Dostępny w Internecie: http://support.sas.com/documentation/onlinedoc/stat/930/introclus.pdf. [cytowanie na str. 26]

[86] Internetowy podręcznik Statystyki udostępniany przez StatSoft. Dostępny w Internecie: http://www.statsoft.pl/textbook/stathome_stat.html. [cytowanie na str. 32, 33]

[87] Oficjalna dokumentacja platformy programistycznej Qt. Dostępny w Internecie: http://qt-project.org/doc/. [cytowanie na str. 105]

[88] Oficjalna dokumentacja programu Weka. Dostępny w Internecie: http://sourceforge.net/projects/weka/files/documentation/3.7.x/WekaManual-3-7-10.pdf/download. [cytowanie na str. 40, 103]

[89] Oficjalna strona internetowa pakietu Rattle. Dostępny w Internecie: http://rattle.togaware.com/rattle-features.html. [cytowanie na str. 35]

[90] Proximetry. Strona internetowa systemu Airsync – Carrier-Class Wireless Network Management. http://www.proximetry.com/airsync-overview.php. [cytowanie na str. 6]

[91] Strona domowa aplikacji Traceis Data Exploration Studio. Dostępny w Internecie: http://www.makingsenseofdata.com/software/software.html. [cytowanie na str. 109]

Dodatek A

Słownik pojęć

Niniejszy dodatek zawiera definicje najważniejszych pojęć użytych w rozprawie doktorskiej, opracowanych na podstawie Encyklopedii Uczenia Maszynowego [56], książek Eksploracja danych [23] oraz Vademecum teleinformatyka [13].

Adres MAC
: Jest to tzw. sprzętowy adres karty sieciowej (interfejsu), czyli 48-bitowa liczba unikalnie identyfikująca kartę sieciową (interfejs). Możliwa jest jednak podmiana tego adresu przez użytkownika, na poziomie programowym.

Atrybut ciągły
: Atrybut mogący przyjmować nieskończenie wiele wartości.

Atrybut dyskretny
: Atrybut posiadający skończoną (policzalną) liczbę przyjmowanych wartości.

Atrybut ilościowy (numeryczny)
: Atrybut, który przyjmuje wartości numeryczne (którym można nadać rangi i poddać podstawowym operacjom matematycznym jak dodawanie czy dzielenie).

Atrybut kategoryczny (jakościowy)
: Atrybut, który przyjmuje jedynie określone, dyskretne wartości (z zadanego zbioru kategorii). Przykładami atrybutów kategorycznych są Płeć, Wykształcenie, Stan cywilny.

Atrybut porządkowy
: Atrybut kategoryczny, którego wartości wykazują naturalny porządek (np. Wykształcenie).

Atrybut symboliczny (nominalny)
: Atrybut kategoryczny, dla którego nie da się określić relacji porządku między jego wartościami (np. Stan cywilny).

Funkcja oceny
Funkcja mierząca jakość dopasowanego modelu – na ile dobrze pasuje on do zbioru danych.

Kanał (transmisyjny)
Jednokierunkowe połączenie między nadajnikiem i odbiornikiem.

Komórka (sieci telekomunikacyjnej)
Urządzenie nadawczo-odbiorcze, wchodzące w skład stacji bazowej.

Korelacja
Współzależność (wzajemny związek) między dwiema zmiennymi losowymi lub wielkościami ze zbioru danych.

Łącze (komunikacyjne)
Zespół środków technicznych służących do przesyłania binarnych, szeregowych sygnałów danych między dwiema, oddalonymi stacjami sieci teleinformatycznej.

Miara podobieństwa
Funkcja, która jest używana do porównania obiektów danego typu. Argumentami wejściowymi funkcji są zatem dwa obiekty, natomiast wynikiem jest stopień ich podobieństwa (często mieszczący się w przedziale liczbowym od zera do jedynki, gdzie zero oznacza zupełnie niepodobne obiekty, natomiast jedynka sytuację, w której są one identyczne). Podobieństwo jest powiązane z pojęciem odległości, która jest jego przeciwieństwem.

Nadajnik
Urządzenie, które generuje pierwotną informację przekazywaną do odbiornika np. telefon.

Odbiornik
Urządzenie odbierające informacje.

Obiekt izolowany (szum informacyjny)
Obiekt różniący się opisem od (wszystkich) pozostałych obiektów w zbiorze, w takim stopniu, że nie jest on podobny do żadnego z nich. Często występowanie obiektów izolowanych wynika z błędów pomiarowych lub brakujących wartości.

Przełącznik
Urządzenie elektroniczne (lub ogólniej element centrali telefonicznej) umożliwiające zestawianie, rozłączanie i zarządzanie połączeniami logicznymi, przez układy fizyczne.

Przepustowość (kanału)
Zdolność kanału do przenoszenia informacji binarnej tj. ustanowienia, ile bitów danych można przesłać w ciągu jednej sekundy przez dane medium transmisyjne, wyrażana często w bitach na sekundę.

Radiolinia
Zespół urządzeń do nadawania i odbierania transmisji radiowych (np. między stacją bazową i jej kontrolerem).

Stacja bazowa
Urządzenie (składające się z wielu mniejszych) umożliwiające komunikację bezprzewodową między siecią telekomunikacyjną a terminalem ruchomym użytkownika np. telefonem komórkowym.

Stopa błędu
Prawdopodobieństwo wystąpienia przekłamania bitu informacji, w strumieniu przesyłanej informacji.

Telekomunikacja
Termin pochodzenia greckiego, symbolizujący przekazywanie wiadomości na odległość bez precyzyjnego określania sposobu i środków przesyłu. Współcześnie mianem telekomunikacji określa się przesyłanie od nadawcy do odbiorcy głosu, dźwięku, faksów i innych danych multimedialnych w postaci sygnałów analogowych lub cyfrowych.

Trafik
Wielkość natężenia ruchu telefonicznego, umożliwiająca określenie intensywności przepływu danych przez urządzenie telekomunikacyjne. Wielkość natężenia ruchu (wyrażona w erlangach) definiowana jest w stosunku do tego, jaki wprowadza przeprowadzenie jednej rozmowy telefonicznej.

Uczenie nienadzorowane
Dowolny proces uczenia maszynowego, którego celem jest poznanie struktury i zależności występujących w zbiorze danych, bez dostarczania żadnych zewnętrznych informacji na temat jego poprawności.

Wzorce (modele)
Zależności i podsumowania będące wynikiem eksploracji danych. Są to przykładowo reguły asocjacyjne, grafy i skupienia. Model najczęściej ma charakter globalny (czyli podsumowuje cały zbiór danych), natomiast wzorzec może mieć zasięg lokalny (czyli może odnosić się wyłącznie do wybranego podzbioru obiektów).

Zbiór danych
Zbiór pomiarów pobranych z określonego środowiska lub procesu (najczęściej powiązany z kolekcją obiektów, dla których zostały one przeprowadzone).

Złożona baza wiedzy
Przez złożoną bazę wiedzy autor rozumie zbiór danych złożonych, reprezentujących specjalistyczną wiedzę dziedzinową.

Dodatek B

Wykaz przeprowadzonych badań dodatkowych

Autor w ramach rozprawy przeprowadził następujące dodatkowe eksperymenty oraz badania (nieomawiane szczegółowo w rozprawie):

- analiza wpływu metody tworzenia reprezentantów skupień na efektywność przeszukiwania takiej struktury (wyniki opublikowano w [68, 72]),
- analiza możliwości wykorzystania algorytmu OPTICS w procesie grupowania i wizualizacji struktury skupień dla danych rzeczywistych (wyniki opublikowano w [73, 69]),
- przegląd literaturowy podejść stosowanych do grupowania dużych wolumenów danych, ze szczególnym uwzględnieniem ich wad i zalet, jak również wykorzystujących je algorytmów analizy skupień (wyniki opublikowano w [48]),
- przegląd literaturowy graficznych metod reprezentacji danych, w tym struktury skupień za pomocą m.in. techniki map prostokątów (wyniki opublikowano w [49]),
- przedstawienie koncepcji grupowania hybrydowego wykorzystującego reprezentantów skupień (utworzonych przez zastosowanie algorytmu gętościowego) do stworzenia dwupoziomowej, hierarchicznej struktury grup (wyniki opublikowano w [74]),
- porównanie efektywności zastosowania hierarchicznych, niehierarchicznych oraz gęstościowych algorytmów grupowania w procesie przeszukiwania danych jakościowych (wyniki opublikowano w [46, 45]).

Spis rysunków

Spis tabel